DESIGN OF

A MANUFACTURING INFORMATION SYSTEM

FOR AN ASSEMBLE-TO-ORDER ENVIRONMENT

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

Capt. A V Ramesh

to the

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1994.

POST GRADUATE OFFICE
I.I.T. Kanpur.
Submitted on 1.4.94

C E R T I F I C A T E

It is certified that the work contained in the thesis entitled " Design of a Manufacturing Information System for an Assemble-to-Order Environment", has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Kripa Shanker)
Professor
Industrial & Management Engineering
Indian Institute of Technology
Kanpur - 208 016

April, 1994

ABSTRACT

This work involves the design of a manufacturing information system for an assemble-to-order, manufacture cum overhaul environment, applying the concepts of Relational Database Management Systems.

The concept of generic, modular bill of material and configuration control in an assemble-to-order manufacturing environment have been implemented using relational database design. The various modules provided includes amongst others, query on features of the various models of the equipment, that are manufactured, selection of an equipment with a given list of features and generating product structure of the end equipment or any assembly or sub-assembly (single-level, indented BOM, summarized explosion and single-level, indented implosion) . These facilities have also been included for the assemblies and subassemblies that go into the manufacture of the equipment. In addition to these, the information system assists in finding out the list of requirements for any assembly that may come for overhaul, once its repair category is known. The information system has been designed such that it can be adapted with little or no changes for implementing any of the following, MPS, MRP, Inventory management, or Maintenance Management.

The environment for which the design has been implemented is an assemble-to-order one where, apart from routine manufacture, overhaul of equipment manufactured earlier is done. Some typical real life examples are heavy engineering equipment manufacturers, railway workshops, and army base workshops. The specific example of an army base workshop assembling and overhauling combat vehicles has been chosen due to the complexity of the structure of the end equipment and also because of the magnitude of the inventory that goes into the final equipment (Approx 60,000-70,000 items). The end equipment here is a conglomeration of communication equipment, vision and range devices,and floatation kits, apart from the usual functional systems and is

available in a number of configurations,thus necessitating a formal information systems to assist in selecting and manufacturing a model, as well as for the efficient overhaul management of the equipment.The implementation has been done using Dbase IV version 2.2 on a Personal Computer.

ACKNOWLEDGEMENTS

1. I take this opportunity to express my deep sense of gratitude and sincere thanks to my thesis supervisor, Dr Kripa Shanker, for his inspiring guidance, constructive criticisms, and constant motivation, without which, this work would probably, not have taken its present shape.

2. The assistance and advise rendered by Shiva, Subranshu, Reddy, Manish, Sanjay, and Sandhya towards the implementation needs special mention. I thank all of them.

3. I shall fondly remember my classmates Arun, Ashwini, Major, Rajesh, Raman, Sharat, Swami, and Yashveer for having made my stay at IIT a memorable and cherished one.

4. I wish to place on record my appreciation for the wonderful faculty of IME Dept, who went all out in providing us with the right environment.

5. I owe a great deal to my wife for putting up with those long periods of absence from home and for encouraging me all along.

Capt A V Ramesh

CONTENTS

APPENDICES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In an assemble-to-order environment manufacturing,each customer order is unique. For every customer order, the functional specifications which define a product that suits customer needs have to be determined. Market trends forces all kinds of manufacturers nowadays to offer a variety of products. The need for such a diversity in final products is strongly felt by marketing and sales. For material management and production , however the commonality in these products is important because a higher commonality improves manufacturing efficiency. To bridge the gap it is customary to derive many different versions from existing products. Therefore many firms deal with range of similar final products and their bill of material(BOM) instead of dealing with single final products.

A thorough understanding of the underlying issues regarding the assemble-to-order manufacturing environment is essential before we proceed to the system design aspects of the manufacturing information system. This chapter deals with the issues that are specific to an assemble-to-order environment. The importance of information for a formal manufacturing information system as also the need for a such a formal information system, are discussed.

1.1 Assemble-to-order Manufacturing Environment

In an assemble-to-order environment the end equipment is a conglomeration or the putting together of various assemblies, with the choice of the individual model of the assemblies themselves being dependant upon the desired features of the end equipment. The final assembly of the various assemblies into the end equipment itself being based on a Final Assembly Schedule(**FAS**). The purpose of the FAS is to plan and control the final assembly of manufactured end items. The FAS for end items represents a

firm commitment to the production of specific end items in the FAS. The typical assemble-to-order firm is also characterised by an endless number of end item possibilities but each of them can be assembled from standard components and options.

At some point the final assemblies are committed to a specific configuration. This point marks the beginning of the final assembly schedule. The length of time associated with the FAS is firm specific. The beginning of the FAS can be thought of as a time fence. To the left of FAS, it is possible to accept orders specifying the configurations which should be assembled during the FAS horizon. Components and major subassemblies have not yet been committed to individual end item configurations. Before this fence, components and major subassemblies have the potential of being assembled into any configuration allowed by their physical characteristics. To the right of the FAS time fence, this flexibility is lost. When production progresses into the FAS by the very nature of being in final assembly, the components and major subassemblies are committed to being assembled into a now specific end item. The potential to be assembled into any end item is now spent on being assembled into a specific end item. The FAS time fence marks a point of no free return. Once the FAS is begun, any work done on the work-in-process inventory must be undone if the parts are to be assembled into a different end item than that specified at the beginning of commitment to well defined end items. The commitment is frozen during the FAS.

1.1.1 Order Acceptance Ratio

Assemble-to-order environments may be further classified by the ratio of time between order acceptance and the beginning of the FAS, and the production lead time less FAS time. This is referred to as the acceptance ratio. If this ratio is near unity, meaning orders are accepted just as production is commencing, then the environment is nearly 'build to order'. The firm needs only to forecast raw material requirements or a few long lead time components in anticipation of future orders. Orders

are received before any significant subassembly schedules are committed. A firm in this position must forecast raw materials usage but does not require accurate forecasting of subassembly demand.

For example consider a tractor manufacturing firm with an eight week production lead time where the final week constitutes the FAS.(Refer Fig. 1.1) During the final week specific tractor configurations are assembled from numerous options. One such tractor may have a medium size engine with standard transmission, standard frame, and a bulldozer blade. Once the tractor enters the eighth week of production the engine, transmission, frame, and blade are committed to the configuration mentioned.

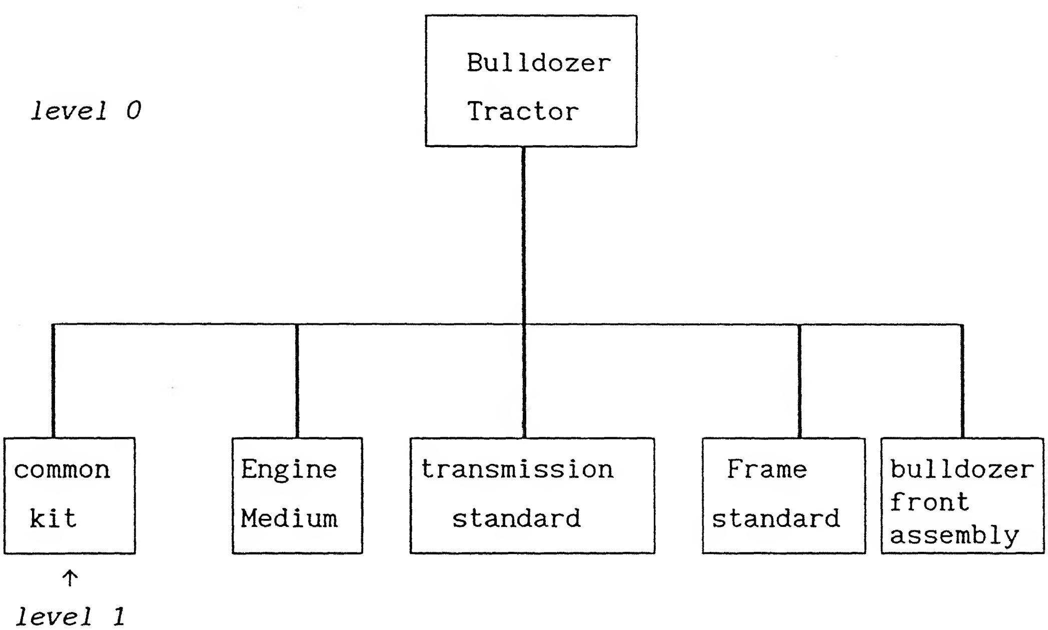


Fig. 1.1 Final week configuration for a bulldozer tractor firm.

feature, that is two possibilities for the eleven characteristics, then the number would double to 19,531,250 = $5^{10} \times 2$.

Even for the tractor company, forecasting at the end item level is precarious. If the company were to sell 100 tractors from each production lot, then a history of sales in the 100 units range would support accurate forecasting of tractor sales at the aggregate level. But in order to support standard Master production schedule(**MPS**), 24 = 3 × 4 × 2 separate forecasts would need to be made. Of the hundred tractors, some configurations will be more popular than others implying that some of the forecasts will be supported by a history of only one or two units each period. This sparse history is inappropriate for accurate forecasting.

The above leads to the observation that accurate forecasting at the end item level for assemble-to-order firms is complicated by two problems of size. First, there may be far too many end item possibilities to realistically consider forecasting. Second, even if the number of forecasts are considered to be manageable, the size of individual histories may be too small to allow the forecasts to be accurate.

We have shown that forecasting the demand for every possible end item configuration is an infeasible approach to constructing an MPS for assemble-to-order firms having a combinatorially large set of possible items. Since the combinatorics of assemble-to-order environments often lead to an unmanageable number of end item configurations , it is similarly infeasible to believe that an MPS system can be based upon individual bills of material for every end item, since the number of such end item bills grow geometrically.

In the assemble-to-order environment one critical managerial issue is length of delivery lead time that must be quoted to a prospective customer. It is realistic to assume that most assemble-to-order firms compete with firms producing similar

products. In order to effectively compete with firms, a firm must be able to at least match delivery lead times quoted by the competition. Competing firms may manufacture nearly identical products. Prospective customers may then choose the firms that can more quickly supply the product. *Delivery lead time performance is the competitive edge of the winning firm.*

As the order acceptance ratio approaches zero, the only way the firm can offer competitive delivery lead time is to carry safety stocks in the various optional features from which customers may choose to order.

For a given investment in safety stocks (over planned or overstated option alternatives) one way to view the managerial problem is, how to construct the MPS in order to maximize the firm's ability to respond to delivery lead time pressures. The manager is interested in the delivery lead time performances that result from alternative techniques for constructing the MPS, and in the comparative costs of designing and maintaining these MPS systems.

It is apparent from the above discussion that setting the MPS for the assemble-to-order environment is a frequent problem. The standard materials requirement planning(MRP) solution of basing the MPS on all end items, each with its own bill of materials, is infeasible in practice, since the combinatorics of some environments produce a prohibitively large number of end items. The MPS must be developed in such a way that does not require an individual forecast for every possible end item configuration.

1.1.3 Alternative MPS Procedures

Of the various techniques available for MPS for an assemble-to-order environment, two alternative MPS procedures that can be easily adopted to the assemble-to-order manufacturing environment need special mention. These procedures are known as the the superbill and covering set approaches.(King and Benton 1988) [12].

Superbills

The superbill approach is in direct response to the large number of end items that need to be handled if the MPS is based upon the standard end item indented bill structure. The superbill allows the MPS for the assemble-to-order environment to be stated in terms of a much smaller number of MPS units. The superbill approach requires the development of special planning bills of material to be used for constructing the MPS. Special bill structures, different from indented bill structures, are designed, maintained and used for setting the MPS.(refer fig. 1.2)

In setting the MPS, someone in the company must commit as to how many end equipments in superbill, or overall unit terms, are to be produced in each time bucket of the planning horizon. This aggregate total may be the production plan, company game plan, or contract between the various functional areas of the firm. Marketing agrees to sell that many overall units; production commits to make them preserving as much flexibility in end item configuration as possible; and finance agrees to provide adequate resources. What is important to superbill development is that the overall or total MPS contains no safety stock or over planning.

The FAS is based upon specific customer orders, although in many firms it is possible for someone (usually marketing) to " write an order on ourselves ". That is, once the production planning commitment is made in terms of overall units, this number, and no more or less, will be built. If insufficient actual customer orders are received between the date of commitment and the FAS, it may be the job of marketing to define the end item configurations as units that will be held as finished goods or perhaps retrofitted later.

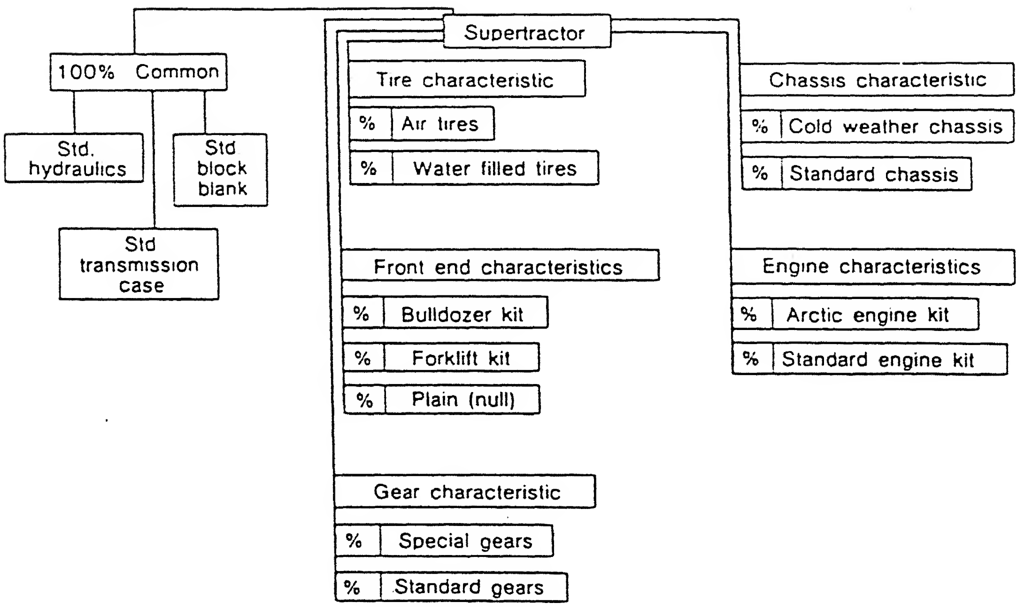


Fig.1.2 An example for superbill

Covering Sets

The main feature of covering set technique is that it enters specific end item (with resultant typical indented bill structures), as opposed to special planning items, into the MPS. These end items can be exploded with standard MRP logic to create component part requirements; the actual end items that are in fact assembled are determined with an FAS that differs from the original MPS. The scheme avoids the need to maintain separate non-indented or superbills while still supplying the arithmetic consistency and percentage mix forecasting as in the case of superbills.

Comparison of Superbill and Covering Sets Approaches

The two MPS techniques, superbills and covering sets, are not necessarily interchangeable. Each technique affects more in firm's operation than just aiding the development of the MPS. The two techniques affect buffering inventories, system development, final assembly schedules, routing files, and delivery quote times in respectively unique ways. Before deciding which one of the two MPS techniques to adopt, one would be interested in the performance characteristics of each technique under different levels of environmental factors such as the level of safety stock investment and the degree of commonality in product bill structures. Furthermore one would be interested in the costs to design, adopt, and maintain a master scheduling technique.

Many environmental factors potentially affect the lead time performance of the scheduling techniques. The investment in buffering inventories will affect the quoted lead time when demand varies above the forecasted demand (percentage mix). Analogously, the demand pattern experienced by assemble-to-order firms will consume the cycle and buffering inventories. In this respect the demand pattern can affect the available to promise lead time. Moreover the degree of commonality that exists within the bill-of-material structure, the relative cost of common parts (ABC analysis), the total number of end items in the environment,

and the managerial decision in support systems all are environmental factors that might affect the available to promise measure.

1.1.4 Final Assembly Schedule(FAS)

In order to comprehend the essence and the true function of the MPS, a distinction must be drawn between it and the final assembly schedule, FAS. This has been touched upon earlier in connection with other topics, but at this point a more thorough discussion is warranted. The distinction between these two schedules is a source of frequent confusion, because in some cases the schedules, although always different in concept, may be identical in reality; i.e., the final assembly schedule may serve as the master production schedule.

The MPS represents an anticipated build schedule. The FAS is the actual build schedule. The MPS disaggregates the production plans into end items, options, or groups of items, whereas the FAS is the last disaggregation --into exact end item definitions. The distinction is that the MPS generally incorporates forecasts or estimates of actual customer orders in its preparation, with actual orders thereafter imperfectly consuming these forecasts; the FAS represents the last possible adjustment that can be made to the MPS ; therefore , it is advisable to make that adjustment as late as possible. Any unsold items on the FAS will become part of the firm's finished-goods inventory.

The FAS is distinct and separate from the MPS. The distinction is more clearly seen in the assemble-to-order manufacturing environment. There the MPS is typically stated in superbills and options, whereas the FAS must be stated in terms of the exact end item configurations.

In our case the MPS is essentially a procurement, fabrication, and subassembly schedule(or assembly schedule- level one and below). Its function is to provide component availability,

and it may therefore be viewed as a *component availability schedule* . In this context the term *component* means any inventory item below the end product level.

The MPS may said to "produce" the mentioned components in support of the FAS. This is true to the extent that these components are part of the bills of material reflected in the MPS. The exception to this rule are items excluded from the planning bill during the process of modularizing the bill of material.

For assemble-to-order manufacturing and make-to-order firms, end-item bills of materials are not maintained. If the FAS is stated in terms of customer orders, it is essential that these orders be translated into the equivalent of a single-level bill of material; that is, these orders must lead to bill of material explosion for order release, picking, and so on. This easily is accommodated if the customer order is stated in the same modules as the planning bill.For instance in the tractor example discussed earlier this would mean that the customer order is stated in brand name, horse-power, terrain capability, miscellaneous function capability, etc.

1.2 An Assemble-to-Order Manufacturing cum Overhaul Environment

The type of firm examined here carries out a multiplicity of operations. It manufactures the various assemblies and subsequently assembles them into various models of end equipment, and periodically overhauls also, those equipment which were manufactured by it earlier. Some 40 % of its activity falls into the repair/overhaul and refurbish category.

The workshop contains a broad mixture of shops ranging from foundry and blacksmith facilities to communication testing facilities and underwater combat vehicle fording (floatation) test bed. The shops are a mixture of conventional, N/C and CNC machines. The shops are controlled by production engineers responsible for various aspects of the operation - Combat vehicles, Bridging vehicles, recovery vehicles, and general

production. The manufacturing and overhaul activities share some of the facilities.

Some principles that MRP logic in particular does not recognize applies to this situation, they are:

Where an item is repairable it should be repaired in preference to replacing by manufacturing/purchasing a new item.

Repair work implies disassembly as well as assembly. Most MRP logic does not provide facilities to plan disassembly.

The various activities that are performed in example firm are :

1. Production of various assemblies that go into the end equipment which includes:

- Manufacture
- Procurement from outside agencies
- Sub contracting
- Assembling to a schedule

2. Overhaul of equipment, which involves :

- Disassembly
- Repair or replacement
- Reclamation
- Refurbishing
- Assembly

Resources

The resources that are utilized in our example firm may be broadly classified as under:

Those used for assemble-to-order manufacturing only.

Those used for overhauling of equipment only.

Common to all or central resources such as forklift trucks, or battery operated bin trucks.

1.3 The Transition from MRP to MRP II

Materials requirement planning(MRP) was originally seen as a superior method of ordering inventory. As it evolved, its major emphasis shifted to scheduling (Establishing and maintaining valid due dates on orders). Today it has been expanded further into manufacturing resource planning **(MRP II)** to include the effective planning of all resources of a manufacturing organisation. As illustrated in Fig.1.3 manufacturing resource planning is a much more sophisticated system which incorporates information from manufacturing, marketing, engineering, and finance into a total operations plan for the organisation. This evolution of MRP to closed-loop MRP or MRP II results in a single game plan to meet the overall goals of an organisation. This is possible because it ties together strategic, financial, and capacity planning areas.

Thus the term MRP has meant different things to different people at different times .Some think of it as an inventory system, others as a scheduling system, and still others as a complete closed-loop production system. It can be all of these things, depending on the organisation and the stages of its development with MRP. Most would agree that MRP fosters systems thinking and tends to become the cornerstone of the production system. Within the limits of its methodology, it will reveal (1) what is needed, (2) how many are needed, (3) when they will be needed, and (4) when they should be ordered.

The time horizon in MRP is composed of equal time periods called " **time buckets.**" The **"time buckets"** are usually weeks or some other convenient time increment. The time horizon is usually longer than the longest sequence of component lead times of any product. It should be long enough to obtain all materials and produce all components before a planned order release for end items. It is also possible to have " **bucketless** MRP ". Where equal time periods are not used but specific dates are developed for every order.

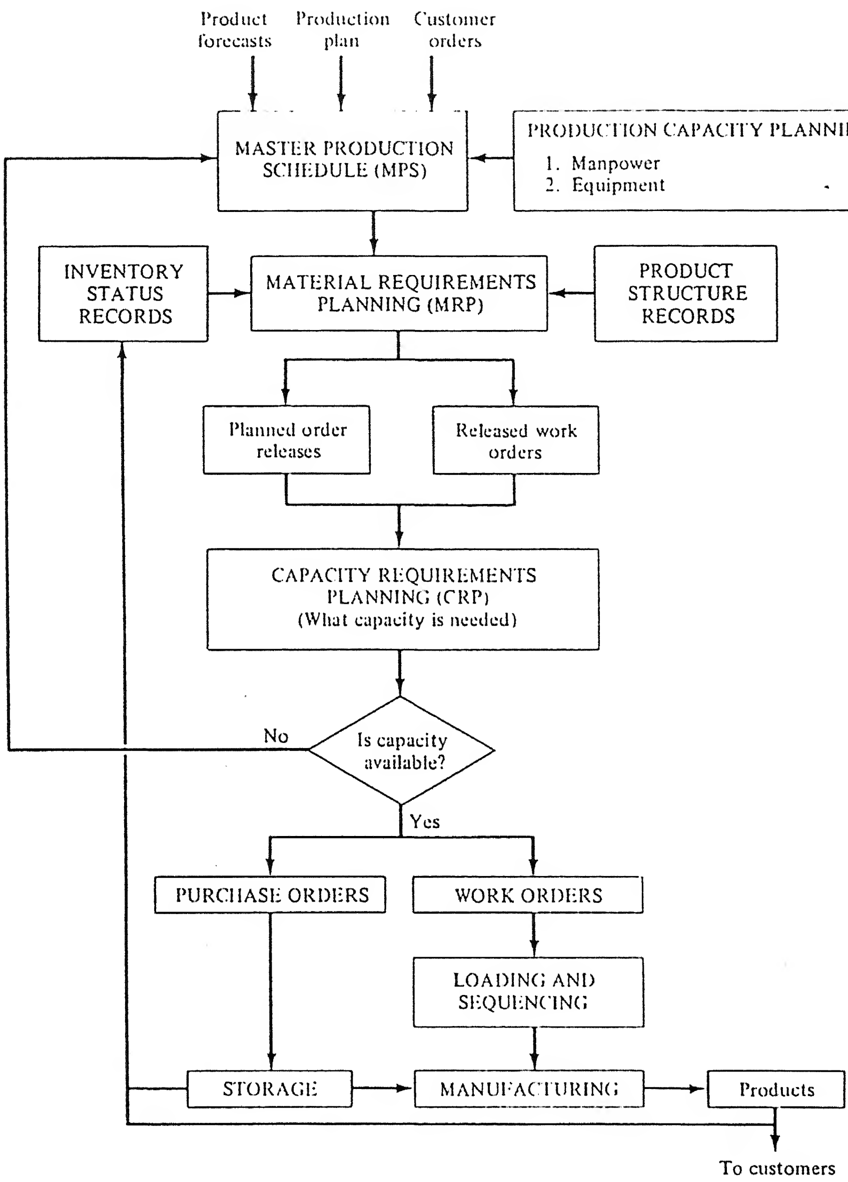


Fig.1.3 A closed loop MRP system

The effective operation and efficiency of an MRP system depends on the integrity of the files and records of relevant data. The quality is directly influenced by data accessibility, up-to-dateness, and accuracy. Lack of record integrity is a major reason for the failure of MRP systems to live up to expectations. Computer based MRP, even more than manual, will not perform satisfactorily with poor files and records. File integrity is not a one-time affair, but must have constant maintenance. The outputs from a computer based MRP system cannot be better than its inputs.

When inventory decisions cannot be separated from production decisions, they must be considered part of aggregate planning for the total production system. Dependent demand inventory used in this category, since they are production dependent. The function of an MRP system is to translate the overall plan of production (MPS) into detailed component requirements and orders. It determines what is to be manufactured and when, as well as what is to be procured and when. For end items, it is useful to hold extra inventory to provide for customer service. To hold extra inventory of components with a dependent demand normally serves no function. While the demand for end items may be uncertain, the demand for components is certain (deterministic) and dictated by the production schedule.

When the following conditions hold, MRP is usually superior to other inventory systems:

1. The final product is complex and contains several other items.

2. The specific demand for the product in ant time period is known.

3. The final product is expensive.

4. The demand for an item is tied in a predictable fashion to the demand for other items.

5. The forces creating the demand in one time period are distinguishable from those in other time periods.

The technique called materials requirement planning (MRP) was

developed specifically for the control of the work and purchase orders for an assembled product. The requirement for each part is calculated according to the demand for the higher level parts in which it is used ,going up the structure until the end products are reached. Their demand in turn is quantified in the final schedule. An MRP extracts from the engineering records the information required to establish deterministic links between the various intermediate requirements, and it thus significantly increases the usefulness of the data supplied by the engineering and purchase departments. In practice, it is software translator for high level requirements that takes into account both current inventory levels and orders in progress to compute net requirements along the entire eligible time horizon.

After solving the problem of order control through MRP the next issue to be faced is the most effective use of production capacity. For the same it is necessary to take the work order backlog, inclusive of the new MRP suggestions, and project the resulting work load for each work center. The software that performs this definitive analysis is the capacity requirement planning(CRP) module. The manpower or machine hours needed for the planned workload are compared with available capacity, along the same horizon used for materials planning. For each lathe and each assembly bench, CRP identifies the periods of saturation and idleness without modifying the priority sequence, but merely highlighting the consequence of the plans already existing. The decision on what to do in case of overload is left to a human planner.

The availability of a complete picture of resources and not just of local evidence of overload allows the experts to advance or retard less critical operations, with the final result of leveling out the original profile.

It often happens, for example, that a temporary constraint escapes notice in the analysis of macroresources so that an apparently feasible final schedule may need some readjustment before it can actually implemented. If this limiting factor is always the same, it can be included among the resources to be kept

under closest control so that subsequent schedules will be more reliable right from the start. CRP thus acts as a bottleneck detector, particularly in environments where manufacturing information systems are being installed to make up for incomplete understanding of the modifications caused by capacity additions or organizational restructuring.

The reassignment of orders ensuing from the overload of some centers has direct consequences on the final schedule. In practice, it becomes necessary to repeat MRP to phase replenishments with the new work_order sequence and this, in turn, makes it necessary to reschedule some shipments for finished products. This chain reaction justifies the name of closed-loop planning given to the complete procedure cycle: each decision triggers another decision which in turn, requires an adjustment to be made on another level.

The three software modules MPS, MRP, and CRP must therefore be incorporated into a broader framework, where they perform integrated functions, even though in apparently remote domains. APICS(American Production And Inventory Control Society) theory unifies them in a single category covering all aspects of manufacturing management. The term that is used to concisely identify all tools applicable to these problems is MRP II or Manufacturing resource Planning.

1.4 Information System in an MRP II System

Having seen the transition from MRP to MRP II it becomes imperative to see the salient role played by information in any MRP II implementation system, before we proceed to see what a manufacturing information system should essentially include in an assemble-to-order manufacturing environment.

The successful implementation of MRP II software systems has provoked considerable interest during the last few years. During the past ten years MRP II has developed into a highly effective management planning and control system encompassing all functions within the manufacturing environment. Advances in computer

architecture have made MRP II affordable even for the smallest of companies, permitting an industry wide application of proven planning and control tools for productivity, quality, and excellence.

Despite the rewards MRP II offers, few organisations have successfully implemented full systems ; in fact the implementation ratio has been starkly disappointing if one goes by the record. There are many reason for MRP II implementation failure and a whole literature has developed outlining pitfalls and describing successful methodologies.

MRP II is direct result of what Oliver Wight confidently proclaimed over twenty years ago to be a revolution in production and inventory management- the ability of the computer to generate information to develop " plans that other people could be held responsible for executing ".(Wight,O,W 1974) [24]. The key word is, of course, "information". What traditionally inhibited effective priority and capacity planning and control had been the inability of statistical and scientific methods to provide timely, accurate, and intelligible information on the current status of production processes.

As it has developed MRP II bridges this gap. As a management tool, a fully implemented and well managed MRP II system provides information that enables functional integration, is interactive, and permits performance measurement and critical analysis. Finally an MRP II system generates information that decision makers can develop into plans coordinating and directing the activities of the whole company to meet a set of common goals. What manufacturing companies are really looking for when they decide to implement an MRP II system is a philosophy of managing information. In reality, there are three categories of information, varying in complexity and efficacy, yet possessing common and reciprocal elements. They are (1) Static data-base information (2) relational information (3) Dynamic information. The path to MRP II excellence lies in understanding and utilizing the integrated nature of these three.

1.4.1 Static Database Information

Static data-base information is the easiest of the three categories to comprehend. All MRP II implementations usually begin by collecting and entering data-base elements such as the charts of accounts, parts master, BOM, routings and customer and vendor records into the system.

The collection, entry, and maintenance of static data-base information is one of the primary on-going tasks involved in implementing an MRP II system. Unfortunately, there is an inclination on the part of management to underestimate both the criticalness and requirements of static data maintenance. The initial obstacle is normally the poor condition of manually maintained or partially automated data. An implementation invariably requires that static files be overhauled and restructured to meet the needs of the new system. File cleanup, however, is only part of the problem. Data-base integrity tends to deteriorate following loading, if a thorough program of file maintenance is not in place.

1.4.2 Relational Information

The most significant attribute of a data-base is that it constitutes a common source of information that system applications can use to add, combine, or relate other information. Much like the human understanding, which uses words to create sentences and paragraphs whose meanings are greater than the meanings of the sum of the individual words themselves. Obviously, relational information is much more complex than static data base information, hence poses new problems.

There are two classes of relational information. The first arises from the use of compiling verbs in sorting, selecting, counting, or listing data-base information. This class can take the form of a report that combines nothing but static data, such as listing of customer number, address, and discount. It could

also take the form of report that associates together static data with interactive data, such as a sort of the parts file by part number and inventory at a specified numerical value. Some examples of the first class of relational information are stock status, MPS, MRP, or open sales order.

The second class of relational information is interactive information. This class of information attempts in MRP II systems to integrate decisions among, and to pass serially information between, departmental functions and subfunctions. When a sales order is shipped for example a number of data files are automatically updated depending upon system parameters. The sales order file, sales order inquiry, shipments record, finished goods inventory file, accounts receivables, and receivables inquiry are all sequentially updated. In addition, forms and reports such as packing slips, bills of lading, invoices and shipping and invoice registers are automatically generated.

The management of relational information is critical to the on-going success of an MRP II implementation. Since relational information can interactively alter the data values of files linked by system programs, requirements that data be accurate, timely, representative of operational realities, visible to users, accessible, and linked logically to other required data values must be scrupulously maintained.

1.4.3 Dynamic Information

The maintenance and utilization of dynamic information poses the greatest challenge to an MRP II implementation. Dynamic information arises from the flow of production activity that occurs on the shop floor. The reason for implementation failure on the shop floor resides in the difficulty MRP II systems have in processing dynamic information. Shop floor information changes almost as immediately as it is established. Purchased material that does not arrive on time or is the incorrect engineering revision; inaccuracies in inventory or BOM records; expediting of rushed customer orders and the corresponding rescheduling of other

orders and their components; machines that break that down; tooling that is defective; absenteeism; scrap; strikes - these and a host of other problems continuously affect shop-floor information.

1.5 Manufacturing Information Systems

Robotics, flexible manufacturing systems, and automated factories are all topics of great significance to the industrial community and are the focus points of the international debate on the economic and social role of the business enterprise. The attention of top management has been drawn back to manufacturing, after a period dominated by concern for marketing and financial issues, in response to the dominant trends of the economic scenario or of the current doctrine. The reason for this renewed interest is in the progress of information technology, thanks to which even the most daring projects become feasible, losses and waste can be eliminated and existing assets are freed for new investment.

Information is today's key resource, and it alone can definitively dispel that fear of excess capacity that many experts recognize as a constant of economic development. Without accurate information even the most sophisticated plants can only provide a greater quantity of the product least required, at the wrong time.

Collecting and classifying production data is useful task in itself and is well within the reach of even a medium sized business. Leaving aside ambitious projects that require high levels of investment, the reorganisation of procedures and the redefinition of data paths yield immediate benefits and open up future opportunities for any company. The main instrument of this strategy is a manufacturing information systems built around a model that reproduces both product structure and process logic. Such an instrument can integrate all factory resources within a single framework that constitutes the necessary reference for manufacturing planning and control. Its functions should ideally

support inventory control, order entry, shop floor operations, while dedicated tasks compute replenishment schedules, balance the daily mix of production and communicate with the intelligent machinery installed in production departments and work centers.

1.6 Need for a Manufacturing Information System

Even in the most prosperous periods, the inventory control function of a factory can manifest symptoms of organisational sclerosis that undermine the efficiency of the various individual units and end up by jeopardizing the effectiveness of the business as a whole. Generally speaking, older firms tend to get into trouble first, but none are completely immune, not even the most recently established . Operating methods and procedures may become outmoded, or be unable to stand the pace imposed by the market. It is these parasitical phenomena of an organisational nature that waste many resources and constitute the real constraints limiting the growth of a business. They are the underlying cause of the law of decreasing marginal returns which is one of the cornerstones of microeconomy, from a certain point onwards, using the same technology, a greater input of a given production factor produces a proportionally diminishing effect on the output. Manufacturing management software, by providing timely warning signals and optimized operational indications, can therefore perform a structural role in the economy of a business and defer the moment when the inversion point of production capacity is reached, thus increasing production without absorbing greater resources.

1.7 Scope of the Thesis

Manufacturing information systems are often perceived as a series of tools for making high-level decisions, carrying out simulations, and optimizing factory operations. Such a conception is reductive, but nonetheless captures the essence of initial contributions these systems made to management. Historically, software came into manufacturing to improve inventory control through the use of advanced purchasing policies, whose parameters were adjusted dynamically using static or iterative algorithms.

Manufacturing information systems are not designed solely for planning purposes, although in this area they achieved their first positive results . Recent systems also include procedural functions to exchange data and validations between different department of the same firm. These functions control the actual execution of planned activities and are the necessary complement of scheduling functions, the more so as they precede them in the implementation of the system. In order to play their roles effectively, the procedural functions must be able to monitor the production process and guide it along its way with timely microadjustments.

The modules of an MRP system constitute a complete but abstract implementation of a total planning philosophy because they do not take into account the specific characteristics of a particular business, as determined by the interaction of market conditions and strategic choices. Field of activity and management style can both affect the relative importance of the various software modules, often making some of them useless. The analysis of industrial typology therefore represents the other side of manufacturing information systems and is the necessary complement of any methodology designed to solve any real-life problems.The comparison of different requirements helps us to bring into focus the critical parameters of each and to achieve a better understanding of the mechanics of the underlying method. The final form that a manufacturing information system may take will ofcourse depend on the individual firm for which it is being developed.

Taking these factors into consideration and the specific example firm that we have chosen , an assemble-to-order manufacturing cum overhaul, the main objective of the present thesis is to design a Relational Database management systems based Manufacturing Information Systems capable of achieving the following features:

(a) Model selection for manufacturing or assembling , and

configuration control.

(b) A generic and modular BOM, capable of generating product structure explosion and implosion, capable of serving to the needs of both manufacturing and overhauling activities.

(c) Maintain and integrate with the main database, a database of all the equipment manufactured till date and in service to assist in drawing schedules for overhaul of equipment,and also to enable redeployment of equipment whenever a need arises for the same.

(d) Maintain and integrate with the main database, a database for the complete inventory ,one that can later be enhanced to do the the following:

MPS

MRP

Inventory order management.

(e) Maintain and integrate with the main database a database to assist in the overhaul of equipment or assemblies.

1.8 Organisation of the Thesis

Chapter II presents the significance of BOM design in manufacturing information system and discusses the merits and demerits of special BOMs that are relevant to assemble-to-order manufacturing environment and goes onto explain the concept of of modular BOM, generic BOM and configuration control.

Chapter III deals with the system design aspects, since database design is one of the key building block of any manufacturing information system. The emphasis being on description , design, and development of databases.

chapter IV describes the implementation details. This is explained using menu structures. Conclusions and the scope for further enhancements and improvements are drawn in chapter V.

CHAPTER II

BILL OF MATERIAL DESIGN AND CONFIGURATION CONTROL

A BOM is a list of items, ingredients, or materials needed to produce a parent item, end item, or product. It can take several different forms and be used in many ways.The choice of the type of BOM design to be used , will greatly depend on the environment of operation of a manufacturing firm. This in turn has an overbearing influence on the choice of software to process the bill of material, for the purposes of MPS and MRP, and the storage methodology that one would like to adopt. This chapter examines the various special BOMs that are relevant to an assemble-to-order manufacturing environment, their relative merits and demerits, and proceeds to discuss the importance of configuration control

2.1 The Significance of BOM Design

The assemble-to-order manufacturing firm is characterised by an almost limitless number of end-item possibilities made from combinations of basic components and sub assemblies. for example the number of unique options of any major automobile manufacturer runs into thousands.(The General motors of U S A has more than one million end item possibilities). Moreover each new product option offered to the customer tends to double the number of end-item possibilities. What this means is that the MPS unit in the assemble-to-order manufacturing environment cannot be feasibly based on end-items. (This aspect has been illustrated by us with an example in chapter I (1.1.2) under; The issue of forecasting in an assemble-to-order manufacturing environment). Now, defining other units for MPS means creating special bills of materials. In this section we present a few key definitions to clarify what a bill of material is and is not. Thereafter we proceed to discuss the special bill of materials that are relevant to an assemble-to-order manufacturing environment. With this background it is possible to see how MPS takes place in the assemble-to-order manufacturing environment.

Key Definitions

The *bill Of Material* is narrowly considered to be an engineering document that specifies the ingredients or subordinate components required to physically make each part number or assembly. A *Single Level bill Of Material* is comprised of only those subordinate components that are immediately required , not the components *of* the components. An *Indented Bill Of Material* is a listing of components, from the end item all the way down to the raw materials; it does show the components of the components.

The *Bill Of Material Files* are those computer records designed to provide desired output formats. The term *Bill Of material structure* relates to the architecture or overall design for the arrangement of bill of material files. The bill of material structure may be such that all desired output formats or reports can be provided. a *Bill Of Material processor* is a computer software package that organises and maintains linkages in the bill of material files as dictated by the overall architecture (bill of material structure). Most bill of material processors operate using the single-level bill of material and maintaining links or chains between single-level files. It is the bill of material processor that is used to pass the planned orders for a parent part to gross requirements for its components.

The single-level bill and the indented bill are two alternative output formats of the bill of material. Alternative output formats are useful for different purposes. For example, the single level bill supports order launching by providing the data for component availability checking, allocation, and picking. The fully indented bill of material is often used by the industrial engineers to determine how the product is to be physically put together and by accounting for cost implosions. A fundamental rule is that a company should have one , and only one, set of bills of material or product structure records. This set should be maintained as an entity and be so designed that all legitimate users can be satisfied.

The concepts presented in the rest of this chapter present another way of thinking about the bill of material in an

assemble-to-order manufacturing environment. The traditional approach is from an engineer's point of view; that is, the way the product is *built*. The key change required to achieve superior master production scheduling is to include bill of material structures based on the way the product is *sold*. In this way the bill of material can support some critical planning and management activities.

Constructing a bill of material structure or architecture based on how a product is sold, rather than how it is built, offers some important advantages. Achieving them, however, is not without cost. The primary cost is that the resultant bill of materials may no longer relate to the way the product is built. Activities based on that structure (eg.,industrial engineering) will have to be based on some new source of data; that is, if the description of how the parts physically go together is not found in the bill of material, an alternative set of records must be maintained. Providing alternative means to satisfy these needs can be costly in terms of both file creation and maintenance.

2.2 Relevant and Special BOMs

The BOM may be called as the product-structure when it indicates how a product will be produced. There are numerous BOM designs that are available for use by the manufacturing firms. Some of the special BOMs that especially are relevant to an assemble-to-order manufacturing environment, and those that have been incorporated in the design of our system will be elucidated in the following paragraphs.

2.2.1 Modular Bill of Material

A key use of bill of material files is translating the MPS into subordinate component requirements. One bill of material structure or architecture calls for the maintenance of all end-item buildable configurations. This bill of material structure is appropriate for the make-to-stock firm, where the MPS is stated in end items. For each end item, a single-level bill is

maintained, which contains those components that physically go into the item. For an automobile or a railway coach manufacturer with its umpteen end-item possibilities, this bill of material structure is not feasible.

A solution is to establish the MPS at the option or module level. (Refer Fig. 2.1) As indicated , the intent is to state the MPS in units associated with the "waist" of the hourglass. This necessitates that bill of material files be structured accordingly; that is, the option or module will be defined fully in the bill of material files, as a single-level bill of material. Thus ,the modular bill of material structure has an architecture that links component parts to options, but it does not link either options or components to end-item configurations. If the options are simply buildable assemblies, then all that is required for the new architecture is to treat the assemblies as end items; that is, designate them as level zero, instead of level one. In most cases, however, the options are not stated as buildable assemblies but as options that provide some services to the customer.

Consider for example the air-conditioning option for an automobile. The single-level bill of material would show this option or module as consisting of a particular radiator, fan, hoses, compressor, and interior knobs and levers. These items are not, however, assembled together. They are assembled with still other parts as assemblies, which eventually are assembled into the automobile.

The use of the air-conditioning option as a bill of material will pass demand from the customer who wants this option down to the necessary parts. It can also be used to forecast demand for air conditioners. however this bill of material is not useful in the physical building process for air-conditioners. For example,

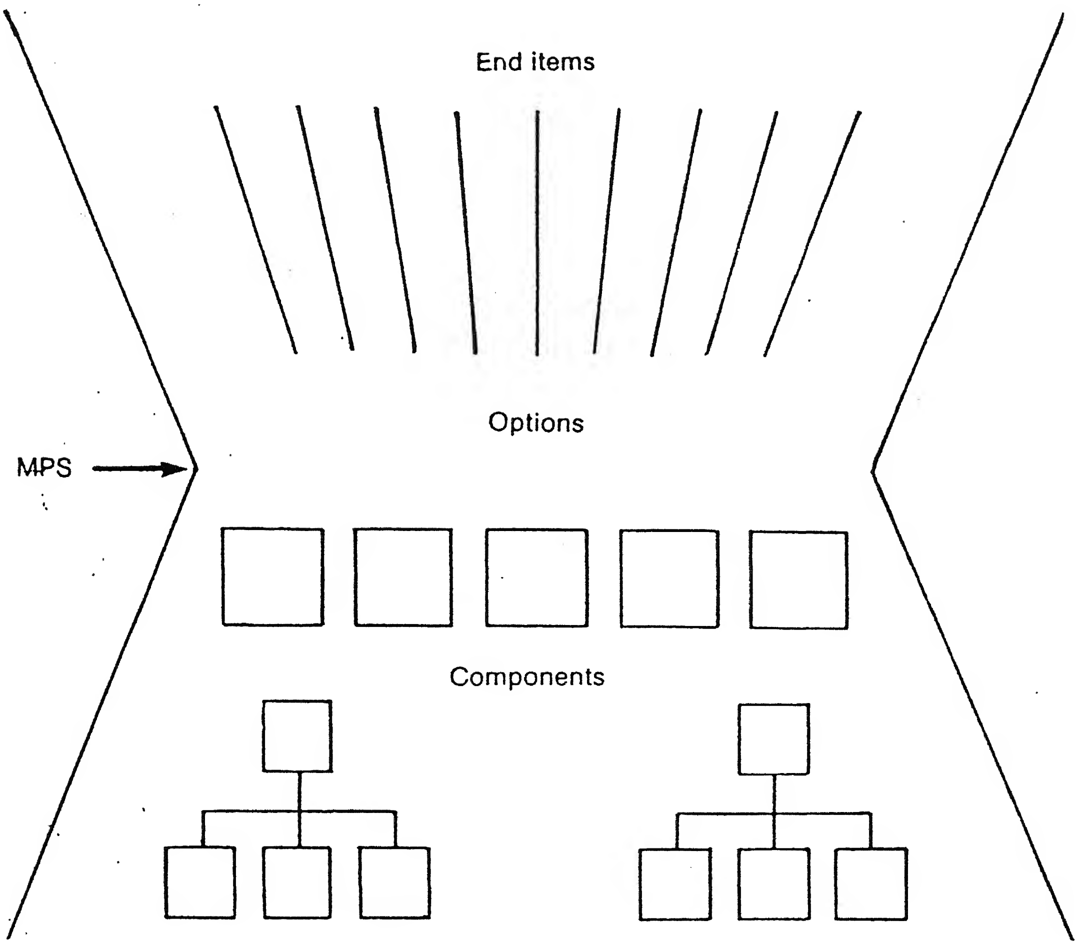


Fig.2.1 Choosing the MPS level

the air-conditioning knobs are planned by the bill of the dashboard assembly where they are installed. Thus the industrial engineer needs other means to say how the dashboard is to be assembled and from what components.

Using the modular bill of material structure the MPS is stated in the terms in which it is *sold*, rather than in terms in which it is built. The approach is compatible with marketing perceptions of models, options, and trends in options. This tends to improve forecasting. The master scheduling task may be made easier by using modular bills, but order entry tasks are made more complex since each option must be evaluated.

Once the individual customer order(representing a unique collection of options) is entered, it serves the function of a one-time, unique single-level bill of material; that is, which options or modules are to be included for the particular customer order. It is controlled by a separate final assembly schedule(FAS).

Modularization can achieve two different purposes;

1. to disentangle combinations of optional product features.
2. to segregate common parts from unique parts.

An example will illustrate the concept of modularity (refer figure 2.2). Suppose a manufacturer offers his customers 10 engines, 30 colors, 4 bodies, and 2 frames. By assembling the optional features in various combinations, it is possible to build (10) (30) (4) (2) = 2400 models or unique configurations.

It would be irrational to set up separate bills for each end product (level-0); 2400 would be needed. Furthermore the development of a MPS would be as illustrated in chapter I, an arduous task. The solution is to disregard specific end item forecasts and proceed with a single forecast for the product family. from it forecasts for each option can be derived from popularity percentages, defined as the probability of option selection based on past customer orders or expectations. these

will make it possible to decompose the family forecast into the forecasts for each option. for example past sales may indicate that 75% of orders call for frame A and 25% call for frame B . If the forecast is for 100 products per period, orders for 75 A frames and 25 B frames would be scheduled. In using the modular method, each of the options or modules would have a BOM, but there would be a total of 10+30+4+2 = 46 bills instead of 2400.

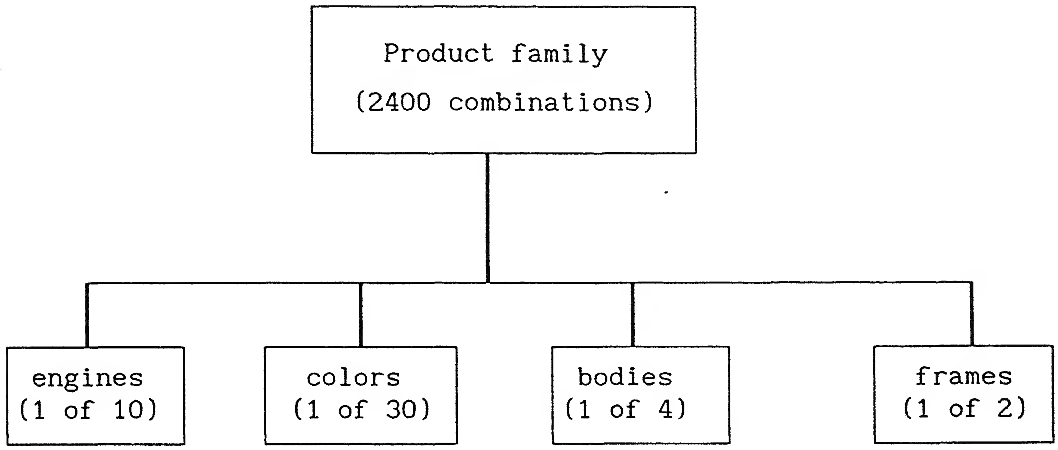


Fig.2.2 Example to demonstrate modular bill of materials

2.2.2 K-Bill

The other terms for K-bill in industrial use are *kit-number* and *K-number*. This is infact a variation of the *pseudo-bill*. In an assemble-to-order manufacturing environment of the kind that we are interested in , for the building or assembling of the end product there is a requirement for a large number of small loose parts such as nuts, bolts and fasteners, normally at level-0, 1, and 2 i.e., end-equipment, assembly, subassembly levels. These loose parts are used to assemble the major assembly units or the subassembly units together. Under an MRP system, to deal with such items individually on the master production schedule level would not be practical. These parts are therefore put into an imaginary bag, as it were, and a part number assigned to identify this bag,

or kit. A pseudo bill of material is established for the kit, which is then treated as an assembly, for purposes of master scheduling and MRP.

The principle involved here is the same as in the case of the S-bill i.e., assigning a single new identity code to individually coded items that constitute a logical group, and employing the format of bill of material to relate such items to one another. K-numbers may be used to advantage within a modular bill (to streamline material requisitioning, for instance), or they may be used even when there is no need for modular bill of material. The K-number is another nonengineering part number, these artificial identity codes have little to do with the design of the product and are not part of product specifications, but are created for more convenient forecasting, planning, and master scheduling. These newly created bills of material, along with the modular bills discussed earlier, represent a superstructure in the bill of material file which once established, must then be maintained along with the rest of the files. This is a new function which increases the cost of maintenance.

2.2.3 Generic Bill of Material

The core of this concept is, that a range of similar products can be described implicitly with one all-embracing BOM instead of enumerating the products individually. An accurate BOM for every single product out of that range can be derived from it. It is essential that each of these products or assemblies can be uniquely identified by a number of characteristics(e.g.,options).

We have seen earlier that for an assemble-to-order manufacturing environment, it is best that the MPS is defined at the assembly level (level-1 ,or a level below the level of the end equipment). This however creates two problems:

1. Forecasting of requirements for MPS.
2. The loss of information about the manufacture of the final product or end equipment.

The generic BOM is a solution for the second problem. A generic BOM can be considered as the opposite of a specific BOM.

Whereas the specific BOM describes exactly one product the generic BOM describes a range (genus) of products. It can be regarded as the all-embracing BOM of all single products within a product family. It can therefore not directly be used for planning or manufacturing purposes. It is only a framework for creating a specific BOM at the time one is needed.

In a multi-level BOM if an assembly or final product contains one generic item at any lower level in its BOM, all intermediate subassemblies containing this generic item are generic too. To generate a specific BOM when a customer order for a specific product has been accepted every generic item in the generic BOM must be evaluated against the options chosen and be substituted by the proper specific item. A specific combination of chosen options may have an impact on many different generic items on different levels of the BOM. The substitution of generic items by specific items only depends on the options chosen.

2.3 Configuration Control

It is necessary to define an additional function to the generic BOM concept to make sure that while specifying or choosing a version of an equipment one ends up choosing a makeable model of the end equipment. The joint choice of two particular versions of two features may actually be mutually exclusive in that such an end equipment with the stated versions of the two features may not be makeable at all. The mechanism that prevents the choice of two or more mutually conflicting versions of the feature of an equipment is called *"Configuration Control"*.The problem of configuration control can be very significant especially in complex situations such as an assemble-to-order manufacturing environment.

In practice fully modular BOMs are exceptional. Also the situation of independent features is very rare. The selection of options is mostly subject to all kind of very complex constraints. In some situations product families have more than 100 features, subject to a number of constraints which is a multiple. The

selection of a valid set of options, identifying a makeable product requires considerable expertise. In these situations there is a large risk of specifying final products that cannot, or may not be manufactured. To prevent manufacturing or purchasing actions to be started for such a final product, every customer order has to be screened. This often a bottleneck in the order entry procedure, sometimes slowing it down by many weeks.

This causes some typical problems such as correctness of information, non availability of information, and finally the accessibility of the information to all the concerned people such as marketing, sales, and order screeners.

The use of configuration control function should guarantee that only products which can be manufactured can be defined. The configuration control contains the knowledge about the dependencies between features and options. The dependencies may be of two types namely:

1. If a certain option has been selected another feature is no longer relevant.

2. If for one feature a certain option has been selected, the set of permitted options of another feature is restricted.(for e.g., if the terrain capability of a combat vehicle has been chosen as desert then the floatation capability is restricted to either shallow or medium fording only).

There are two extreme ways to use a configuration control function. In between these two extremes there are shades of combinations.

1. The configuration control system does not support the end-user in determining an initial set of options. The end-user must define this set using his own knowledge. Afterwards the configuration control system is used to check for conflicting options.

2. The end-user is guided through the process of configuration. he must interactively select options. Every time he selects an option, the configuration control system executes the consequences. Features and options which are no longer relevant or

permitted are not even shown. Using this approach, an end-user can never define an invalid set of optional values.

We intend using the first approach since the information systems is primarily for the more experienced users of the firm, and is to serve to select a model of an equipment, if available, once the options for the various features are specified.

2.4 Practical Concerns in Implementation

It needs to be stressed here that implementing the modular bill, K-number, configuration control and generic BOM is not an easy job. While the cost of classifying and maintaining information increases in establishing a modular bill and K-number, the addition of configuration control further increases the complexity of the information system. The configuration control determines the diversity in final products. it is of utmost importance that:

(a) a suitable set of features is chosen to characterize products;

(b) the permitted values these features may take is specified at the time of selection of an equipment or assembly or a subassembly as the case may be; and

(c) interdependencies between choice of different options of the different features is recorded,

These decisions affect many manufacturing functions. There must be an unanimous agreement between the people from various departments as to the diversity of the final products.

CHAPTER III

SYSTEM DESIGN

A robust system design is the basic foundation on which any information system may generally be built upon. The robustness itself being dependant upon the choice of an appropriate data model, file structures that are capable of capturing the complete information with the least amount of redundancy, and the flexibility of the design to be able to adapt to different applications with as minimum change as possible. In this chapter the issues of, choice of data model, the suitability of the relational model, and the design framework of the database that is proposed to be used for the information system are elaborated.

3.1 The Role of Database Management in Manufacturing

Given that dynamic changes occurs constantly in the physical flow of material in any manufacturing firm there is a need to maintain accurate information in the manufacturing information system database. To do so will require high quality information. This means that the database elements must reflect the physical reality. Only by keeping the data in the system accurate can the outputs of a system be truly integrated into day-to-day decision making. The salient points that need to be kept in mind while designing a database for manufacturing applications are:

The database must reflect reality
Database transactions must be processed rapidly
Database maintenance must be tightly controlled
Users actions must be integrated with database transactions
The system must tell the truth to the users
The informal system must die.

Proper maintenance of manufacturing information system database requires three distinct integrated efforts by any organisation for its successful implementation. First there is a set of technical efforts to ensure proper backup of data files, integration among files and consistent system uses of the files.

The second set of efforts is on the part of system users to ensure that database needs become an integral part of their daily working lives. The third is to make a deliberate effort to base functional planning and decision making on the one integrated database.

Like all other corporate resources information is a critical resource and must be managed and controlled for effective decision making. Databases as a repository for stored data which is integrated and shared by different users forms an important ingredient of any information system.(Date,C.J) [8]. With this as the background, we now proceed to see what a relational database is, its relative merits and demerits vis-a-vis others, its flexibility for use in manufacturing applications and the important role it plays in the design of a manufacturing information system.

3.2 Relational Databases

In principle a relational model views data in the form of a table having a fixed number of columns and rows. Moreover it is that ability to analyse the functional dependencies among the fields (columns) that makes the relational data model powerful, through a series of procedures called normalization, the functional dependencies are analysed and unwanted dependencies are reduced or eliminated. A table that has no undesirable dependencies is said to be devoid of any update anomalies. An update anomaly arises if an insertion, deletion or change of the content of the table causes inconsistencies in the data.

The operation and manipulation of tables can be expressed algebraically in terms of three basic functions: selection, projection, and join. Selection creates subsets of all table records, projection creates subsets of all the columns in a table, and join allows combining of tables.

3.2.1 Advantages over the Network and Hierarchical Models

Relations are easy to understand. The number of basic data constructs is one, namely the relation or table itself; all information in the database is represented using this one construct, and moreover this one construct is both simple and highly familiar. As for keeping distinct concepts separate, there seems to be few, if any instances of "bundling" in the relational approach. Indeed it is significant that most of the research into such areas as concurrency, locking, security, integrity, view definition and so on have taken the relational approach as a starting point precisely because it provides a clean conceptual base. The normalization discipline guarantees that the same fact will not appear at two places.

Relations are also easy to manipulate. This statement is true at both the tuple at-a-time and set-at-a-time levels; in other words, very high-level operators are available, as well as the more familiar low-level operators. The number of distinct operators in any given language is small because there is only one type of data construct to deal with; essentially we need just one operator for each of the four basic functions retrieve, insert, delete, and update. Unlike the hierarchical or network data model, the relational model does not make use of any explicit paths to indicate the inter-record associations. Additionally, relational models treat data retrieval on a table basis and not on a record basis. These two characteristics, however imply that database design will be effective only if the designer has a comprehensive and intense understanding of the data relationships.

An MRP system is data driven and additionally, there is a set of complex, recursive data relationships. While network data models could handle recursive relationships with design modification, the relational data model offers the best implementation. The recursive relationships have to be decomposed first and the tables normalized. The other inherent advantages of a soundly designed relational database are :

1. A high degree of data independence

1. A high degree of data independence

2. It provides a community view of the data and in spartan simplicity, so that a wide variety of users in an enterprise(ranging from the most computer naive to the most computer-sophisticated) can interact with a common view.

3.2.2 Flexibility in Manufacturing Applications

Relational database management systems are an efficient new tool to assist in managing businesses and processes, with properties that allow not only planned, prespecified, reports and queries of the database but also unplanned, ad hoc accesses.The capability of a relational database management systems(RDBMS)includes amongst others:

Search the data on a variety of conditions

Sort output on various attributes

Modify the structure of the data and data types

Add new data, columns, or tables

Perform simple string operations and substring functions on alphanumeric data

Classify data into groups

Dynamically create indices on attributes to increase the efficiency of certain operations

Validate data entry according to type, domain an combination.

Thus a relational database provides both power and flexibility to customize applications to the needs of the users. The relational database also provides the facility to adapt to new needs that may develop. Modifications can be made to the application as well as the structure of the data with minimal changes to existing applications. These features make the relational database an invaluable tool for a variety of manufacturing activities. The maintenance of data integrity by minimizing data duplication and the interaction of data to join pieces of information for management or reporting purposes(since they have been split apart for data integrity) are features that find wide application in manufacturing. The areas of maintenance management, purchasing and inventory management, group technology,

and master scheduling find most application in a manufacturing environment.

3.3 Classification and Grouping Scheme Adopted

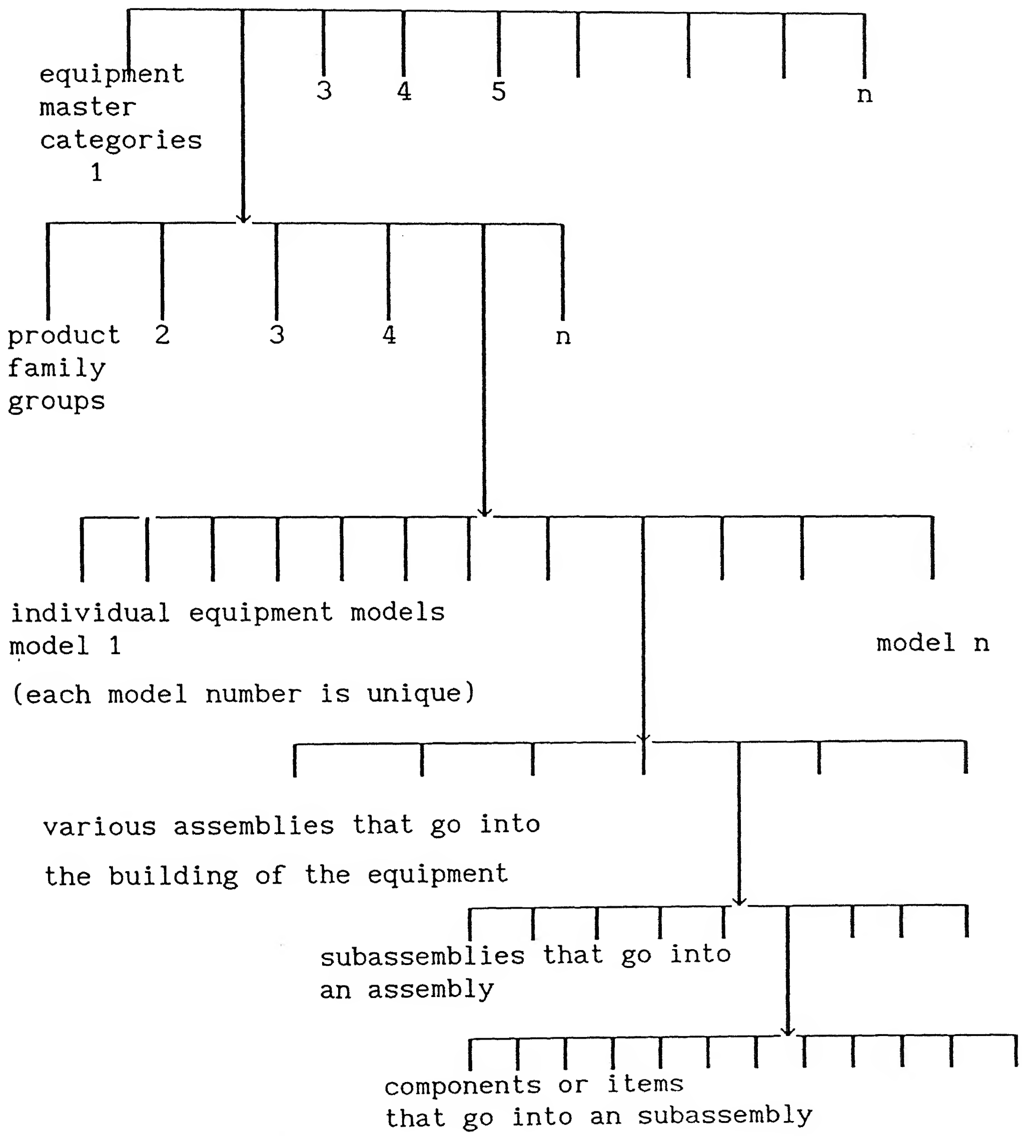
For convenience of implementation and also for easy maintenance of the databases that we are to design, a classification and grouping scheme for the example that we have chosen, a manufacture cum overhaul assemble-to-order environment where combat vehicles assembly and overhaul is being done, is proposed in the following paragraphs.

3.3.1 End-Equipment

In order to design a database for a manufacturing information system it is necessary to understand the environment thoroughly. The firm that we are talking about is involved both in the manufacture as well as the overhaul of the equipment. The equipment themselves fall into one of the master categories. Each master category may in turn have more than one product family groups, and each of these product family groups of a master category may have a multitude of models or options available within themselves (refer fig. 3.1). Each of these models of a product family group of a category is built using the required model or version of the various assemblies that go into the assembling of the final equipment of that model.

A typical commonplace example is that of an automobile manufacturing firm (except that they do not carry out the overhaul of the automobiles). Where the master categories themselves would be the sports/racing category, limousine category, family car category, and the station wagon category. The product family groups inside a master category would be, for example in the

racing category, be horse-power rating(600hp, 650hp and so on). The models inside a product family group of a master category would be color options, number of gears, body type, size of tyres and so on. With the total number of makeable combinations in such a firm running into hundreds if not thousands, there is definite need to evolve a proper classification scheme.



note: These items may themselves be the result of assembly of other items and may have more than one level in the product structure but are referred to as items in general.

Fig.3.1 End-equipment categorisation scheme

3.3.2 Assemblies and Subassemblies

By referring to the end-equipment as level-0 the assemblies that we are talking about are those building blocks or modules at level-1 that together make the end-equipment. By subassemblies we are referring to those modules or building blocks at level-2 which assembled together results in an assembly of level-1.

Continuing with our example of an automobile manufacturing firm the assemblies here would be the engines, gear boxes, air-conditioners, suspension systems, superstructure, etc.. Each of these assemblies is available in a number of models. for eg., the engine assembly may be available in different horse-power ratings, different cooling systems or lubrication systems adopted, different configurations for the same horse-power output, such as V-type construction (to reduce the overall height, which is a requirement for racing cars), or inline construction. The sub assemblies here would mean the fuel-injection-pumps and the air-compressors of the engine, shock absorbers of the suspension system, or the gear-shifting mechanism of the gear box assembly, amongst others.

One more point that needs to be emphasised is that the assemblies broadly classified, fall into one of the two following groups:

1. **Common to all** These assemblies are used by all categories the equipment. For example the same model of engine assembly may be used for a limousine as well as a station wagon or for that matter a family car. This is generally true with most of the assemblies in an assemble-to-order manufacturing environment.

2. **Specific** These assemblies are specific to one master category or at times, one product family group of a master category and are not used in other master categories. For example the air-conditioner assembly may be specific to limousine category only, the supercharger may be specific to the racing car category only.

3.3.3 Items

By items we are referring to every single component of the inventory. The assemblies and subassemblies discussed earlier also form an effective member of the inventory. Being an assemble-to-order manufacturing firm and also because the MPS is done at level-1 instead of level-0 (for the reasons brought out earlier in chapter I of this thesis) every component; assembly, subassembly, or item from level-1 and below constitute the inventory. Since the total number of items may be around 60-70,000 in such a firm it would be prudent and convenient if one were to classify them based on their type and where-used data, so as to divide them into similar and easily manageable groups. This is to facilitate the search, locate, and update operations that are so frequently required in a manufacturing information systems of the kind that we are interested in.

One such illustrative grouping scheme for the automobile firm example would be; the prime-mover group(for all items that go into the engine, gear box, suspension system, etc.), the controls and instrumentation group(for all dashboard instruments and gadgets) miscellaneous group(one for internal and one for external, essentially for the various fitments and accessories which do not necessarily form a part of any of the functional groups). The suggested grouping scheme is by no means exhaustive and may actually vary from firm to firm, based on its individual range of products and needs.

3.3.4 Coding Scheme Adopted

Technically speaking, the process defines the product, taking into account the implementation technology of each step together with any internal or external interface devices. From an information-control standpoint, the coding conventions divide the process into two integrated aspects: structure and routing. These

far reaching implications on the two main concepts of manufacturing management underline the importance of choices affecting part numbers. The coding and grouping scheme for the equipment and the components of inventory that we propose to adopt is as shown in fig. 3.2 .

The coding system is flexible enough to accommodate future changes, additions and deletions. It takes into account the complete gamut of inventory database and all applications where a database, based on such a coding scheme may be used.

INDIVIDUAL EQUIPMENT NUMBER

1	2	3	4	5	6	7	8	9	0

			range
1 & 2	--	Year of manufacture	
3	--	equipmentcategory identity	(00 - 99)
4	--	Product family group identity	(0 - 9)
5 & 6	--	model identity within a PFG	(00 - 99)
7 to 10	--	unique individual serial number of equipment	(0000 -- 9999)

Fig.3.2 Coding scheme

(contd. on the next page)

INDIVIDUAL ITEM NUMBER

1	2	3	4	5	6

			Range
1 & 2	--	Alphabetic item class identifier	(aa -- zz)
3	--	Item master group identifier	(0 -- 9)
4 to 6	--	individual unique serial number of item within a class	(000 -- 999)

For e.g., item master groups in an automobile firm may be:

0	--	Prime mover group
1	--	controls and instrumentation group
2	--	miscellaneous group (internal only) etc.

Item classes may be:

eg	--	engine	bo	--	bolt
gb	--	gear box	st	--	strip steel
nu	--	nut	wh	--	washer , etc.

Fig.3.2 Coding scheme

3.4 Design : Product Structure

Product technology, choice of components and processing methods are all strategic resources in a modern company. They are often the result of extensive research. The computer files containing the details on all these aspects constitute the nucleus around which the manufacturing information system is built. The bills of materials for each product may be considered the crucial point of contact between technical and business applications and represent a permanent reference for the various transactions. Operating procedures, planning systems and cost calculations are all based on the contents of the product files, where ease of access is essential.

In designing the database, the aim is to effectively represent the environment by an entity-relationship (E-R)diagram. The E-R diagram will essentially identify the entities or objects present in the system, and show how the entities are related to each other. Once the objects, their attributes and the relationships are identified and depicted in the form of an E-R diagram, the diagram can be mapped onto a set of database files. By implementing the database files on the computer, the environment can be effectively captured and stored in the database.(Nandakumar,G.,1990) [14].

In the case of the bills of material processor(BOMP) the E-R diagram is being created primarily to capture the parent-component relationship shown partially in fig. 3.3(a). Only two objects are needed to describe the BOMP environment. They were namely the assembly and component objects.At some point, an assembly could also appear as a component of a different assembly. The two objects thus defined were related to each other by a many-to-many relationship. The many-to-many relationship was derived from the fact that one item may be used in more than one assembly and one assembly may contain more than one item. Fig.3.3(b) shows the E-R diagram representing the BOMP environment.

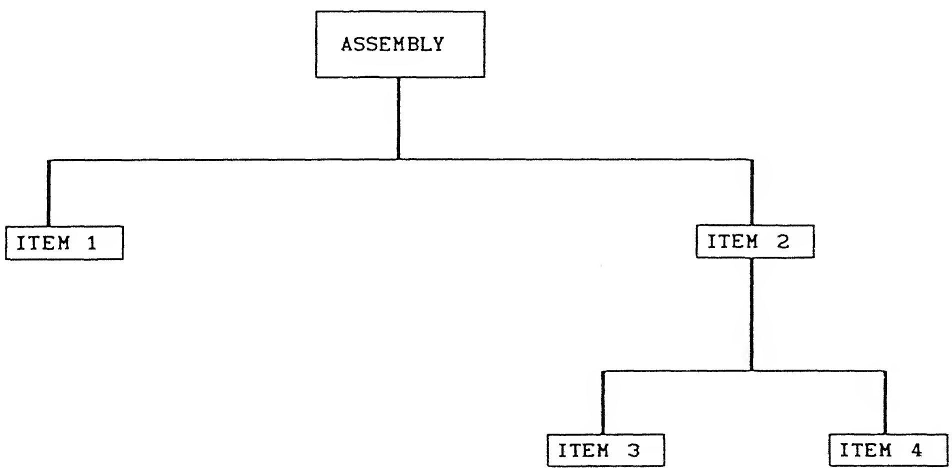


Fig 3.3(a) Example to show Parent-component relationship

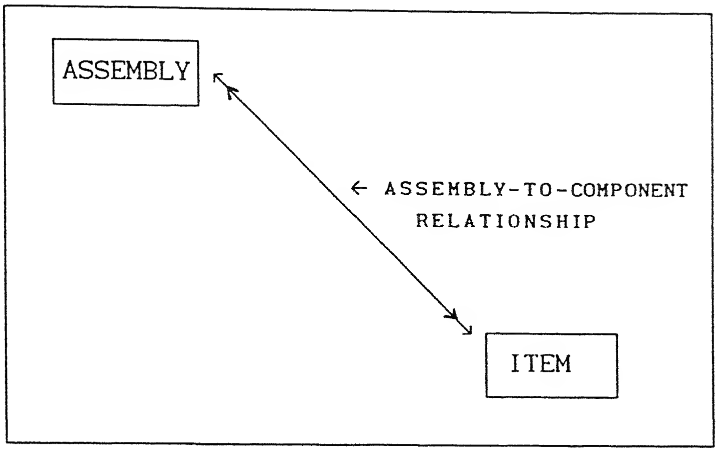


Fig.3.3(b) Assembly-item relationship

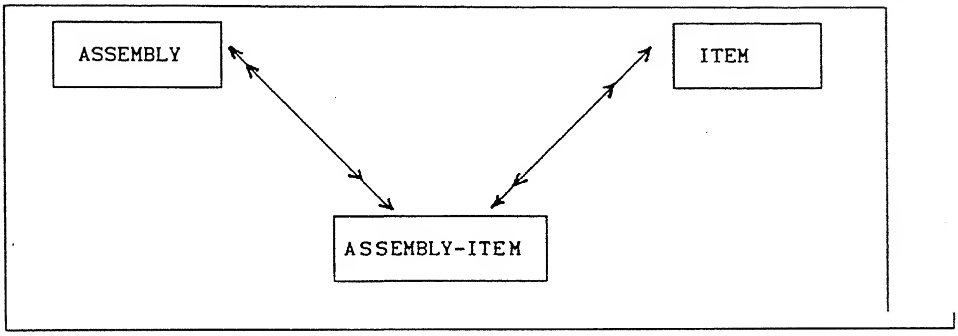


Fig.3.4(a) An expanded relationship for implementation

```
Item_master   (item_no,item_description,safety_stock,...)

Assembly_item    (item_no, parent_no, quantity)
```

Fig.3.4(b) File structure adopted for implementation

According to relational database theory, a many-to-many relationship can be implemented on the computer only after it is broken down into two one-to-many relationships. Fig.3.4 (a) shows

the two one-to-many relationships created to replace the many-to-many relationship. In theory, generation of this E-R diagram marked the end of the database modelling. Because of the fact that the relational data model can readily be expanded in a modular form, other objects could also be brought into the picture in order to broaden the scope of the database. (Nandakumar,G.,1990) [14]. for instance the information about suppliers could be incorporated in the database by defining a new object called *vendors*.

Bill of Material Storage Methodology

Under manual methods, product-structure data can be maintained in a variety of formats, and it is not unusual for different departments in a plant to maintain their own versions of the BOM, in a format to suit their respective needs. Thus an engineering bill, a purchasing bill, a costing bill, an assembly bill, and an inventory planning bill may exist simultaneously. Needless to say this is a costly and inefficient way to maintain product-structure data files.(Orlicky, J., 1975) [15].

When product-structure data are stored in a computer system, duplication of effort(and cost) in maintaining the bill of material can be eliminated. Only one set of data, in a format best suitable for storage, is maintained, but the data can be retrieved and "assembled" by the computer in a variety of formats to be printed out (or otherwise displayed) for the benefit of the various users.

The format used for the purpose of storage is the so-called *single-level bill*, which minimizes both storage space and maintenance requirements. Under this approach, a single product -structure record for each assembly and subassembly is established, and it simply lists the parent item's components(including the quantity per information) on the immediately lower level only. Multi-level bills are then reconstructed via chaining as and when required.

The file structure that has been adopted is shown in fig. 3.4(b). For the BOM file the keys that have been used are:

Item within parent - for product structure explosion

Parent within item - for pegging and where-used data.

3.5 Design: Features Management

As discussed in chapter II (refer section 2.4) features management design (or configuration control) is to enable the following:

1. Listing of the features of a model of end-equipment, or assembly, or a subassembly.

2. To assist in checking if a model of the end-equipment or assembly, or subassembly exists, with a given list of features and if yes, list out the model numbers.

The relations for the feature management was planned at three levels i.e., level-0, level-1, and level-2. This decision is purely based on implementation convenience and the applications for which we intend using our database. The decision to have feature management only till subassembly level(The example of a "Fuel injection pump" that goes into an engine assembly as discussed in section 3.2.2 of this chapter is relevant to our discussion here) has been taken , taking into account the conflicting features of increased availability of information, and increasing volume of data and consequently increased cost of maintenance.

In all these relations (level-0, level-1, level-2),the "*entities*" are the various models of the end-equipment, or assembly, or subassembly.The "*attributes*" or the characteristics of the entities are the various features of the end-equipment, or assembly, or subassembly as the case may be. The "*domain*" for each of these attributes, which is a collection of all values that may occur for a specific field type is the permitted values that the attributes may assume. In our case it would also mean stating a makeable or manufacturable option. The keys here would be the

equipment model numbers, or the assembly model numbers, or the subassembly model numbers as the case may be. The file structure is shown in fig. 3.5.

End-equipment -- One Relation for each category
(model_no, feature1, feature2, feature3,...feature n, remarks)

Assembly -- One Relation for each family, eg.,engine
(model_no, feature1, feature2,feature3,...feature n, remarks)

Subassembly -- One Relation for each family, eg., FIP
(model_no, feature1, feature2, feature3,...feature n, remarks)

Key in all relations -- model_number

Fig.3.5 File structure: Features Management

3.6 Design: Inventory Management

The inventory management system database needs to be designed in such a manner so as to enable one to get the following done out of it:

MPS
MRP
Order Management
Purchase Management, to name an important few.

We now proceed to the design considerations and thereafter to the prototypic model for such a database.

3.6.1 Design considerations

For the master production scheduler to operate effectively, it is critical that there be one single unified database for the MPS,

that it link to the production plan and to the MRP, and that clear responsibilities for all transactions be established. This involves not only the usual data integrity issues but also some organisational issues.

In the case of MPS many of the transactions occur in different functional areas. For example the receipts into finished goods may come from completed assemblies (production), the shipments from order closing (marketing) or bills of lading(finance). It is critical that exact responsibilities be established for transaction processing, and that the data linkages to MPS system and files be religiously defined and maintained.(Vollman *et.al.*1989) [23].

Another critical database requirement for the MPS is proper control over both engineering and nonengineering changes to the BOM database. The MPS is often stated in planning units that may not be buildable. This requires a more complex BOM or product-structure database. The result is greater need to procedurally control all changes to the BOM and to evaluate the impact of changes both from engineering point of view and in terms of the effect of nonengineering bills of materials.

3.6.2 Design

A prototypic relational model proposed by Chu and Nilakanta(Chu and Nilakanta, 1988) [6]. for an MRP system is being adopted with necessary changes to suit our environment i.e., assemble-to-order manufacturing. The model is shown in fig. 3.6(a) and the details of the various relations are shown in fig. 3.6(b). In addition to these, the design of such a MRP system must consider several structural and operational features. For instance, a successful implementation must consider the following issues: which update option -- regenerative or net change --to be applied? (Orlicky,J, 1975) [15], what unit of time bucket to be used? How long of the planning horizon is more appropriate? which lot-sizing technique --LFT, POQ, EOQ, PPB etc. -- to be used?

which buffering mechanism -- free stock or safety stock as a trigger system -- to be used? A full understanding of these design issues are essential for the design of the system.

End_equipment-Assembly(equipment_model_No,assy_no1,assy_no2...)
relation

Assembly-subassembly(assembly_model_No,subassy1,subassy2,...)
relation

subassembly-item(subassembly_model_No,item1,item2,item3,...)
relation

Item master relation	(item_no,...)
BOM relation	(item_no, parent_no, quantity)
Source relation	(item_no, vendor_no, unit_rate, supply_lead_time)
Vendor information relation	(vendor_no, city, state, rating street,pin_code)
MPS Relation	(item_no,...)
On Order relation	(item_no,...)

Fig.3.6(a) Prototypic Relational model for MRP

1.	Item master relation		
*	item_no	character	6
	item description	character	30
	low_level_code	character	5
	lot_size	numeric	5
	assembly lead time	numeric	2
	repair lead time	numeric	2
	unit of measurement	character	15
	safety stock	numeric	6
	on hand new	numeric	6
	on hand repairable	numeric	6
	quantity awaiting repair	numeric	6
	unit weight	character	8
	source type	character	15
2.	BOM relation		
*	item	character	6
*	parent	character	6
	quantity	character	5
3.	vendor relation		
*	vendor_Number	character	5
	vendor_name	character	15
	street	character	20
	city	character	10
	state	character	10
	pin_code	character	6
	vendor_rating	character	1

Note: An asterisk indicates that it is the key

Fig.3.6(b) Suggested file structure for the relational model

1. Equipment-constituents(or building modules)Relation

*	Equipment_No	character	10
	assembly 1	character	10
	assembly 2	character	10
	assembly n	character	10

Note: The assembly 1,2,..,.n field contains the model of an assembly that is fitted in the equipment

2. Equipment-Performance History Relation

*	Equipment_No	character	10
	usage criteria 1		
	usage criteria 2		
	usage criteria n		

Note: The field_type and Field_width will depend on the how the individual criteria is expressed. A criteria for example may be "total hours run"

3. overhaul management relation

*	assembly_model_No	character	6
*	repair_category	character	1
	subassembly 1	character	10
	subassembly 2	character	10
	subassembly 3	character	10
	remarks	character	250

Note: The subassembly 1, 2, ...,n field contains the information about the item kit that is required to carry out the overhaul of this subassembly of the assembly for a certain repair category of the assembly. The remarks field is to store general information about the repair to the assy.

Fig.3.7 File structure equipment & overhaul management

3.7 Design: Overhaul and Equipment Management

The firm that we are interested in involved in the manufacture as also the overhaul(repair and refurbishing) of the equipment manufactured earlier. Some typical examples are; a railway workshop, earth-moving equipment manufacturers and army base workshops. These firms manufacture equipment that are highly specialized in nature and due to the prohibitive cost of the equipment, there is a need to enhance the total in-service life of the equipment that are manufactured, by carrying periodic overhaul of the manufactured equipment.. Due to the complexity of the end-equipment and the super-specialization of the various assemblies that go into these equipment, it becomes economically unviable for any agency,other than manufacturer to set up a facility for overhaul of these equipment and the onus of repair and overhaul falls on the original manufacturer only. An example will illustrate this point. The combat vehicle (popularly known as "tank" or "armoured fighting vehicle") has diverse systems such as; laser-range finder, armament system, anti-aircraft system, floatation kits for deep, medium or shallow fording, air-compressors to air start the engines(780-1500hp), hydraulic systems, ammunition storage and loading, auxillary engine to name a few. The facility to overhaul an equipment of this nature will require very high investments, and may not be cost effective at all, thus necessitating the manufacturer to provide for overhaul.

The primary aim of designing a database for overhaul and equipment management is to enable achieve the following:

1. To issue the advisory notice recommending an equipment that is in-service for overhaul once they have attained the performance limits that are specified for them.

2. To find out if an equipment is in-service, that which possesses a given list of features. This is to enable the redeployment of equipment to a place where an equipment possessing certain features may be required.

In order to issue the advisory action of recommending an equipment for overhaul, those that have during their in-service exploitation attained certain limits of performance, it is necessary to maintain and continuously update the database about the equipment that are in-service. The updation itself being done on a periodic basis on receipt of information from the users of the equipment.

To achieve the same we need two databases:

1. **Equipment in-service database** This will have the various equipment that are in service as the entities, the attributes shall be the various performance parameters of the equipment, and the domain for these attributes will be the range of values that a performance parameter of the equipment may take. For example, if the performance parameter is total kilometers run, than the domain shall be all values (usually integer only) from 0 onwards to say 10,000 or as the case may be. The key in these relations shall be the individual equipment numbers.

2. **Equipment constituents database.** Having requisitioned an equipment for overhaul based on its attainment of certain specified limits we would now be interested in knowing what had initially gone into the making of the equipment i.e., the various models of assemblies and subassemblies that have gone into the making of the end-equipment that has now come for overhaul. This we need to know to carry out the repair, refurbishing or replacement to the various assemblies of the equipment. for this our relation will have the various equipment that are in service as the entities, the attributes shall be the various assemblies(or building blocks, or level-1 constituents) and the domain for these shall be the various models of the respective assemblies of which anyone may have gone into the equipment. The file structure of the relations are shown in fig. 3.7.

3.8 Design and Interaction of Databases

For designing the databases the following factors were considered:

1. Attributes or fields
2. Primary key
3. Third normal form
4. Method of searching

The attributes or fields of a relation represent the information that it has to store. The selection of attributes is done with due weightage and redundancy is avoided.

The selection of primary key is a prime factor in the design, since it is an attribute with values that are unique within the relation and can be used to identify the tuples of the relation. To facilitate easy retrieval , editing, and deleting of records from relation the normalization concept is induced while designing. The databases are maintained in the third normal form. The effectiveness of the database mainly depends on how well the data can be retrieved from different relations.

In any database that may be designed to serve a manufacturing information system it is important that the various relations are linked (or related) and also that the linking is explicit enough to permit future users, who may want to enhance the capabilities of the system by the addition of more databases to serve more functions. The interaction of the various databases are as shown in fig. 3.8.

CHAPTER IV

IMPLEMENTATION

In the previous chapter an overview of the designed system and the structure of the various databases under each subsystem had been presented. In order to demonstrate the efficacy and adequacy of such a database we chose to adapt it for a firm which is involved in the manufacture cum overhaul of combat vehicles.

The present software has been developed in the relational database management system, Dbase IV version 2.2 and has been implemented on a Wipro genius 386 machine. The software developed is interactive in nature. The complete software contains mainly, databases, output and report generating programs.

The construction of the software is explained using the menu structure shown in fig. 4.1., which navigates to all the modules of the system.

The main menu screen consists of pads and the navigation of the user to a, pad activates that module and lists out what that module is capable of doing. the user may once again navigate within the module between the various options and decide to choose any one of them. There exists the facility to return to the module to which this option belongs to, or to go to any other module of the main menu, as also to quit from the main menu, and to quit the Dbase session. This facility is available from each and every option inside all the modules. There is also a message prompt at the bottom of the screen which explains what a particular option inside a module is capable of doing. This message changes as one moves from one option to another option within a module or from one module to another within the main menu.

The various databases that are being used are demonstrative of, and conform to the design discussed in the previous chapter. The various file structures and the keys used are illustrated at

MANUFACTURING INFORMATION SYSTEM MAIN MENU

1.	FEATURES SUBMENU
2.	STRUCTURES SUBMENU
3.	EQUIPMENT IN_SERVICE SUBMENU
4.	INVENTORY MANAGEMENT SUBMENU
5.	OVERHAUL MANAGEMENT SUBMENU
6.	EDITING AND UPDATING SUBMENU

FEATURES SUBMENU

1.	EQUIPMENT NUMBER--FEATURES
2.	FEATURES-- EQUIPMENT NUMBER
3.	ASSEMBLY NUMBER--FEATURES
4.	FEATURES--ASSEMBLY NUMBER
5.	SUBASSEMBLY NUMBER--FEATURES
6.	FEATURES--SUBASSEMBLY NUMBER

STRUCTURES SUBMENU

1.	EQUIPMENT MODEL NUMBER--STRUCTURE
2.	ASSEMBLY MODEL NUMBER--STRUCTURE
3.	SUBASSEMBLY MODEL NUMBER--STRUCTURE
4.	ITEM NUMBER --IMPLOSION DETAILS

Fig.4.1 Menu structures showing various possible options

(contd. on the next page)

EQUIPMENT IN_SERVICE SUBMENU

1. EQUIPMENT NUMBER--CONSTITUENTS
2. EQUIPMENT NUMBER--PERFORMANCE HISTORY
3. PERFORMANCE HISTORY--EQUIPMENT NUMBERS
4. EQUIPMENT NUMBER--FEATURES
5. FEATURES--EQUIPMENT NUMBER

INVENTORY MANAGEMENT SUBMENU

1. ITEM AVAILABILITY STATUS
2. ITEM WHERE USED DATA
3. ITEM GENERAL INFORMATION
4. ITEM ORDERING

OVERHAUL MANAGEMENT SUBMENU

1. EQUIPMENT OVERHAUL OPTION
2. ASSEMBLY OVERHAUL OPTION
3. ITEM NUMBER--FOR WHAT REPAIR USED INFORMATION
4. ASSEMBLY & REPAIR CATEGORY--RESOURCES UTILIZED

EDITING OPERATIONS SUBMENU

1. ADDITIONS
2. DELETIONS OR REMOVALS
3. EDITING AND UPDATING

Fig.4.1 Menu Structure showing various possible options inside the modules.

appx"A" to this thesis. The explanatory remarks on the necessity of certain attributes in the case of some files have also been included as foot notes to the respective file structures. The list of the various program files that achieve this implementation and their functions is included as appx "B" to this thesis. The interrelationship between the various program files and the database files have been illustrated at appx "C" to this thesis.

4.1 Features Submenu

This menu has been designed primarily to achieve three tasks, that of; listing out the features of a chosen model of an end-equipment, assembly, or subassembly, listing out the model numbers of end-equipment, assembly, or subassembly given a list of features that one desires, to find out what goes into the making or the product-structure of a so chosen model of end-equipment, assembly, or subassembly. This facility has been extended to assemblies because, being an assemble-to-order manufacturing environment where overhaul also is being done, very often one is on the lookout for a specific model of an assembly, for a particular model of the end-equipment possessing a desired list of features. As was explained earlier in chapter III (section 3.3.2) some of the assemblies are common to more than one category of the end-equipment. By an extension of the same reasoning the facility to select models of subassemblies has also been included.

In the case of selection of a model of an end-equipment, or assembly,or subassembly with a given list of features the user is also prompted the valid options or choices for a feature. So as to assist the user in making a decision. Consequent to selection of a model of an equipment, or assembly, or subassembly with a given list of feature options, the user is given the option of viewing or not viewing the structure of the so chosen model so as to find out what goes into the making of the model. The inclusion of remarks field is to facilitate, the listing of any information about the entity (or in our case the model of the end-equipment, assembly, or subassembly) which could not otherwise possibly be shown under any specific attribute.

4.2 Structures Submenu

This menu is capable of the following:

1. Given an equipment model number, or an assembly model number, or a subassembly model number, lists out:

- the single-level bill of material
- the indented bill of material
- the summarized explosion

The user is given the facility of choosing the category in the case of the end-equipment, the type in the case of subassembly and subassembly. Having decided the category of the end-equipment or the type of assembly or the subassembly as the case may be, the user is is to select the model of the chosen category of end-equipment, or the assembly, or the subassembly, from a scrolling list of model numbers.

2. Given an item it can do its implosion i.e., trace upwards and list out:

- the parents at immediately one level upwards
- the final assemblies i.e., the assemblies at level-1 which require this item for their manufacture.

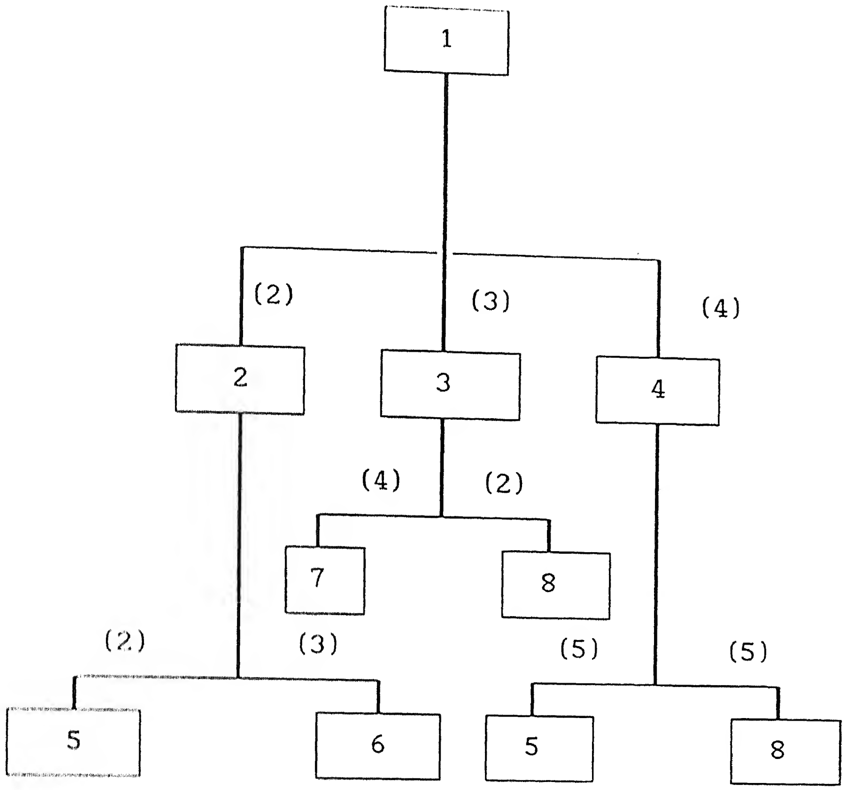
4.2.1 Output Formats of Bill of Material

To understand the above mentioned outputs it is necessary to see the various output formats of bills of materials. An example for each will be in place, for such a discussion. We shall do the same with an example. The structure of our example item "1" is as shown in fig 4.2(a). We shall now see the various display formats for this example.The data can be formatted and displayed in various ways; the six most popular formats are termed as follows:

1. Single-level explosion
2. Indented explosion
3. Summarized explosion
4. Single-level implosion
5. Indented implosion

6. Summarized implosion

The single-level explosion format is that of single-level



Note: The quantities required per are shown in brackets

Fig.4.2(a) Example structure to demonstrate various output formats of bill of materials

Assembly	component part number	quantity per assembly
1	2	2
	3	3
	4	4

Fig.4.2(b) Single-level explosion format

bill, and displays the components used at a specific level of assembly. Several single-level bills are designed to represent completely the product structure of a multi-level product. Refer fig.4.2(b)

The **indented explosion format** lists components on all lower levels and the number of levels involved is indicated in the display by an indentation of component item numbers(or child item numbers) under the respective parent-item number. The indented format represents the product in the manner in which it is manufactured. For our example item "1", the printed output, also known as indented parts list would appear as follows:

Item	Parent	Quantity/parent
2	1	2
3	1	3
4	1	4
5	2	2
6	2	3
7	3	4
8	3	2
5	4	5
8	4	5

Fig.4.2(c) Indented explosion format

The **summarized explosion format** lists all the components of a given model of end-equipment, or assembly, or subassembly, with quantities per reflecting use per unit of the end-equipment, or assembly, or subassembly in question rather than per unit of the item's parent. for our example item 1 the output would appear as follows:

Item	Item Description	Quantity/final assembly
2		2
3		3
4		4
5		24
6		6
7		12
8		26

Fig.4.2(d) **Summarized explosion format**

The item descriptions have been intentionally left out here for simplicity sake but would however form a part of the display otherwise. However, note the quantities against item numbers 5 to 8. The quantity shown here is the sum total of all requirements of the items that are required to assemble item 1.

The **single-level implosion format** is that of a where used list. The output lists all the parents(on the immediately higher level only) of a given item such as, in the case of item "5" of our example the output would be:

component part number	assembly where used	quantity per assembly
5	2	2
	4	5

Fig.4.2(e) Single-level implosion format

The **indented explosion format** traces the usage of a given item in its parent, and in turn the parent's parent etc. , until the end item is reached. Indentations signify levels. The output for our example item "7" would appear as follows:

component part number	assembly used on	quantity per assembly
7	3	4
3	1	3

Fig.4.2(f) Indented implosion format

The **summarized implosion format** is an expanded where-used list in which all items on higher levels that contain the item in question are listed. The quantities per are total quantities of the item used in each of the higher-level assemblies. For our example item "5" the output would appear as follows:

component part number	assembly used on	quantity per assembly
5	1	24
	2	2
	4	5

Fig.4.2(g) Summarized implosion format

The six display formats just described correspond to the six retrieval programs usually supplied by computer manufacturers as part of the BOM processor software. These programs may be used in the process of exploding requirements in an MRP system, preparing material requisitions or picking lists, product costing, engineering changes, etc. To meet special needs however, additional display formats can be created using the same BOM structure that is as of now, available and being used for the

normal formats discussed above.

4.2.2. Algorithm Employed for Explosion and Implosion

The algorithm that is to be used for product-structure explosion and implosion is important in that it should be robust enough to handle the following situations and the output should be factually correct :

1. It should be level independent, i.e., the number of levels below an equipment, or an assembly, or a subassembly, has may be anything from 2 upwards to 100 or even more, and when the program is executed for any of these , it must give correct results and the actual product-structure. Also the level at which an item may occur in a product-structure should have no bearing on the output.

2. In the case of summarized explosion the listing should be; of the total requirements of an item, which means it should have taken into account the level at which the item is required and the quantity, and then multiply it by the quantity required of the parent, by the parent's parent and this continues till the level of the assembly, or the subassembly(depending whether we are doing the summarized explosion for an assembly or a subassembly) has been reached(level-1 and level-2 respectively in our case). This needs to be done with the item at each place it occurs in the structure of the end-equipment, assembly, or subassembly. Subsequently the requirements at various places are added to get the total requirement. The flow charts showing the program flow for indented explosion, summarized explosion and implosion are available as figs. 4.3(a), 4.3(b), 4.3(c) respectively. A sample output of an indented bill of material, and the summarized explosion of a model of an end-equipment are shown at appx "D" to this thesis

step 1. Locate equipment model number in structures database of the equipment category. Collect and store information about models of various assemblies fitted in the equipment.

step 2. Do step3 to step8 while not all assemblies of the equipment are over.

step 3. Locate for assembly model in the respective structures relation of the assembly. Collect and store information about models of subassemblies fitted in the assembly.

step 4. Do step 5 to step8 while not all subassemblies of the assembly are over.

step 5. Locate for subassembly model in the respective subassembly file . Collect and store information about the items that go into the subassembly.

step 6. Do step 7 to step8 while not all items of the subassembly are over.

step 7. Locate for item in the parent field of the respective BOM file.

step 8. Do recursively the following:
locate for each item in the parent field
store and display information about quantity of items (or children)required by the parent
terminating criteria for recursion being when there are no more items below a parent.

note: for equipment explosion -> step1 to step8
for assembly explosion -> step3 to step8
for subassembly explosion -> step5 to step8
the lowest level items have a "$" as their children. This has been done for establishing criteria for termination of recursion.

Fig.4.3(a) Algorithm for indented explosion

step 1. Locate equipment model number in structures database of the equipment category. Collect and store information about models of various assemblies fitted in the equipment.

step 2. Do step3 to step8 while not all assemblies of the equipment are over.

step 3. Locate for assembly model in the respective structures relation of the assembly. Collect and store information about models of subassemblies fitted in the assembly.

step 4. Do step 5 to step8 while not all subassemblies of the assembly are over.

step 5. Locate for subassembly model in the respective subassembly file . Collect and store information about the items that go into the subassembly.

step 6. Do step 7 to step8 while not all items of the subassembly are over.

step 7. Locate for item in the parent field of the respective BOM file.

step 8. Do recursively the following:
locate for each item in the parent field
store and display information about quantity of items (or children)required by the parent
terminating criteria for recursion being when there are no more items below a parent.

note: for equipment explosion -> step1 to step8
for assembly explosion -> step3 to step8
for subassembly explosion -> step5 to step8
the lowest level items have a "$" as their children. This has been done for establishing criteria for termination of recursion.

Fig.4.3(a) Algorithm for indented explosion

step1 To step7 are same as for indented explosion, given on page number 69 of this thesis.

step 8. Do recursively the following:

Locate for item in the parent field collect the information about item (child) and the quantity required by the parent of the child. Now append a blank record in the temporary file and store information about the item and quantity. Before putting the information about the quantity into the temporary file check the requirement of the parent by the parent's parent. If found multiply the quantity with this quantity. This is for level multiplication. After doing this delete the record having the information about the parent's parent since this is no more required.

Where the parent'parent is not found append as it is. Terminating criteria for recursion being when there are no more items below a parent. This is identified by the presence of a "$" in the item field against a parent.

step 9. Open the temporary file. Index this file on the item field sum up the quantities individual item numbers wise. This is necessary because an item may be required by an assembly or subassembly at more than one place in its product structure. List or display in serial order of individual item number.

for	end_equipment	-->step1 to step 9
	assembly	-->step3 to step 9
	subassembly	-->step5 to step 9

Fig.4.3(b) Algorithm for summarized explosion

step 1 Locate for item in item field of respective BOM file. Collect and store information about all its parents and the quantity required of the item by the parents. Store information about the parent and the quantity required in a temporary file.

step 2. If single_level indented implosion is desired then stop else proceed to step3.

step 3. Do recursively the following:

Repeat step 1 with each parent of the item. Subsequently do the same with each parent's parent.

while doing this store particulars of item or in our case the unique item identity number and the quantity of the item required by this parent. Multiplication for levels needs to be done simultaneously. For e.g. , If an item "2" at level 3 requires quantity 10 of item "1" which is at level 4, and item "3" at level 2 requires quantity 10 of item 2(level 3); the total requirement of item"1" for the assembly of item"3" will be 10 * 10 = 100 nos.

The termination criteria for recursion being when there are no more parents or level 1 in our case has been reached by the implosion process.

step4. Use the temporary file and do the following:

If an assembly or subassembly or item requires the inputted item at more than one place add the requirements to get the total requirement. Eliminate repetitions because the tracing upwards would have resulted in the assembly being reached through two different routes since the inputted item would be occurring in the product structure of the assembly at more than one place. Display the indented and summarized implosion information.

note: The highest level item (orassembly), that which does not have any parent above it are identified by their absence from the item field of the respective BOM relation. (the fields in the BOM relation are item, parent, quantity)

Fig.4.3(c) Algorithm for indented implosion

4.3 Inventory Management Submenu

The various options that are available in the inventory management module are discussed in the following paragraphs.

The item availability option of the inventory management submenu lists the availability status of any chosen item. It lists the , quantity on hand new, quantity on hand repaired(now serviceable), quantity on hand repairable(awaiting repair) and the safety stock of the item.

The item used option of the inventory management submenu lists for any item all the assemblies at level-1, their model numbers, their descriptions and the total quantity of this item these assemblies require. It takes into account for, the requirement multiplication by levels and also for the item situation where an item may be required by an assembly(level-1) at more than one place in its structure.A sample output is available at appx "B" to this thesis.

The item information option lists out the complete information about any item that we may be interested in knowing.

The item ordering option is capable of doing; voluntary ordering, periodic ordering.

In voluntary ordering the user is permitted to choose from any of the vendors of the item one desires to order and the quantity to be ordered can be specified by the user, the default option being the lot size of the item. On completion of entry of these, the system generates a report in the name of the chosen vendor, placing an order for the specified quantity of the item.

In periodic ordering, the system lists out all the items whose total quantity on hand has gone below the safety stock, one by one. The user as in the earlier case is given the option of choosing the vendor as also of specifying the quantity required. The default as in the earlier case being the lot size of the item.

The options available under the inventory management module

are by no means complete by themselves. The ones that have been included are more from the point of view of demonstrating the efficacy of the designed database. There are many more functions that one may want to get out of an inventory management system, some of them being, a complete MRP and MPS subsystem, purchase management, and an accounts receivable sub system.

4.4 Equipment Management Submenu

This submenu is capable of achieving the following:

1. The equipment-performance option enables one to find out the performance characteristics of any equipment in service. This is to help one find out the general fitness state of the equipment that are in service. This also assists in future planning for ,or forecast the expected numbers of equipment that may require overhaul.

2. The performance characteristics-equipment option is to enable one to specify the performance limits for a category of equipment and find out how many of the equipment in service have attained these limits. This is primarily to initiate advisory action recommending those equipment that have attained the limits for overhaul. The limits may be the total kilometers run, total hour run, and so on. The user is provided with both the "and" and the "or" options. Also where a limit is not specified a default minimum is assumed. The user is assisted in specifying the limits for the performance characteristics.

3. The equipment number-feature option lists out the features of any equipment that may be in service.

4. The features-equipment number option is to enable to find out if there are equipment in service that possess certain features. This information is to enable one to redeploy the equipment to a place where an equipment possessing certain features is desired. This searches through the database for the equipment number that has features matching to our requirements.

5. The equipment number-constituents option is to enable one to find out what has gone into the making of the equipment. This information is necessary once an equipment reaches for overhaul. At this juncture one needs to know the models of assemblies that

have gone into the equipment that has come for overhaul.

4.5 Overhaul Management Submenu

This module does the following:

1. The assembly repair option firstly gives the user the option to choose a type of assembly and subsequently to select a combination of the assembly model number and the repair category. The system then proceeds to list the summary of requirements of items that may be needed to carry out the necessary repair to the assembly falling under the specified repair category. The listing is similar to the summarized explosion format of BOM discussed under inventory management module.

2. The equipment overhaul option is to know the total requirement of items that may be needed to carry out the complete overhaul of an equipment. The user is first given the option of choosing the category of equipment. Subsequently the repair category for each assembly that goes into the end-equipment needs to be entered. This is because the overhaul of different assemblies are done separately, independent of each other and different assemblies may have different repair categories. Here the output would be a summary of the complete requirement of the items necessary to carry out the end-equipment's overhaul.

3. The item_number where used option is capable of listing the model number of the assembly and for which repair category of the assembly the item is required. This is to enable one to find out the effect of availability or more importantly the non-availability of a particular item on the overhaul operations

4. The resources utilized option is to find out the resource utilization work center wise by an assembly with a certain repair category. This is to assist in the overall resource management of the firm, and to optimally allocate resources between manufacturing and overhaul depending the priorities at the time of allotment.

4.6 Editing operations Submenu

This module demonstrates the standard record addition, record

deletion, record editing and records updating operations of a database. These operations are achieved through the designing of custom screens where the field names are expanded for the benefit of the user, the template and data validation specified such that it is not possible for one to inadvertently enter numeric data into a character field or vice versa. The screen is also presented in a more user friendly "*form format*"as against the usual "*columnar format*". Facilities also exist where the user may be denied the privilege of editing the value of certain important and key fields to avoid data loss by accident.

CHAPTER V

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

5.1 Conclusions

Relational database technology offers firms an opportunity to more thoughtfully and easily examine and process data without direct knowledge of data manipulation or storage. As manufacturing environments become increasingly complex due to government regulations, customer demands for custom built products or features, internal reporting requirements, the need for better management of a firm's database is apparent. Relational database management system software provides users with greater flexibility in selecting through a variety of query specifications, sorting directly on secondary attributes, linking multiple sources of data, and amending existing data as need for new information materialize. It not only simplifies current operations but provides the tool to perform other functions not previously available.

These and other features of a relational database have been utilized in designing a manufacturing information system for an assemble-to-order manufacturing environment. The implementation has thereafter been done for a firm involved in the manufacturing and overhaul of combat vehicles.

The concepts of relational database design have been used for the design of the database, based on which a manufacturing information system can be implemented. The system consists of a variety of databases, so structured and linked to hold the data about; vendor, equipment in-service, features of end-equipment, assemblies, and subassemblies, product structure information or bill of material information of all assemblies and subassemblies that are manufactured, and the data that may be required for overhaul management of assemblies.

The complete transactions have been categorised into modules

and options within modules. In the search for models of end-equipment, assemblies, and subassemblies, satisfying a given set of conditions, the user is assisted by stating alongside a feature, the valid options for that feature. the product-structure explosion and implosion subsystems are robust enough to handle any complicated product-structure, and is level independent..

5.2 Scope for Further Enhancements and Improvements

The modules provided in the implemented manufacturing information system are by no means exhaustive by themselves. There is an immense scope for enhancement and improvements by the addition of more modules or by the addition of more options inside the various modules, as also by the restructuring of the modules themselves. A few such possible enhancements are given in the following paragraphs.

The configuration control that has been used by the system can be best used by experienced end-users only. This can however be further enhanced by the incorporation of a configuration control, where the end-user is guided through the process of configuration(i.e., the selection of end-equipment, assembly, or subassembly possessing a list of features). In this case the users interactively select options. Every time he selects an option, the configuration control executes the consequences and features and options which are no longer relevant or permitted are not even shown. Using this approach, an end-user can never define an invalid set of parameter values(or an unmakeable product).

As was discussed in chapter III under design: inventory management, with the addition of relations relevant to MPS and MRP, such as the MPS relation(item_No, T-bucket, required quantity), On_order relation(item_No, T-bucket, receipt quantity), to name a few, and their linking to existing ones, the manufacturing information system can easily be enhanced to carry out MPS and MRP, once the other issues that have been mentioned in chapter III, section 3.6.2, are resolved, since the item master database and the product structure database already exist. To

support such a master production scheduler , there is the need for the time-phased MPS-record-oriented software system, which must be capable of producing the time-phased records to maintain the database, provide the linkages to other critical systems, provide MPS monitoring and exception messages, and provide for all MPS transactions. Included in this would be, entering of order quantities into the MPS, firm planned order treatment, removing MPS order quantities, changing the latter's timing or amount, converting MPS quantities into final assembly schedule (FAS) quantities, launching final assemblies, monitoring FAS scheduled receipts for timing or quantity changes, closing out FAS receipts into finished-goods inventory, and providing for all customer order entry pegging and promising activities.

The facility to introduce engineering changes and have these changes implemented in the database have not been included and merit consideration. Also the creation of receipts and issue database and their linking with existing databases needs to be done to further enhance the information system.

At the current state of technology these tools, i.e.,the information system, interact with workers and foremen, but in a automated factory or in flexible process, they will interface directly with intelligent machinery. We can anticipate a factory in which the final schedule, analyzed and validated by MRP methods is directly implemented by a network of computers installed in the various departments, each performing a specific part of the centrally formulated program. In such an environment manufacturing information systems will be able to fully develop their potential and really deliver all the benefits promised.

REFERENCES

[1] Alagic, Suad, "Relational database technology", Springer-Verlag, 1990.

[2] Balcerak, K. J. , and Dale, B. G. , "Structuring modular bills of materials with usage pattern analysis", Int. J. Prod. Res. , Vol. 30, No. 2, pp. 283-298, 1992.

[3] Bedworth, D. D. , and Barley, J. E. , "Integrated production control systems", John Wiley & sons, 1982.

[4] Brooks, G. H. , "MRP with a micro computer" , Computers and Industrial Engineering, Vol. 9, No. 2, pp. 297-302, 1989.

[5] Cherian, Mathews, M-tech Thesis, entitled, "Manufacturing planning and control in a make-to-stock environment" IIT, Kanpur 1992.

[6] Chu, Chao-Hsien, and Nilakanta, S. , "On the design of a micro-based MRP system: a relational database approach", Computers and Industrial Engineering, Vol. 15, No. 1-4, pp. 146-152, 1988.

[7] Dangerfield, B. J. , and Morris, J. S. , "Flexibility of relational database for manufacturing applications", Production and inventory management journal-second quarter, 1990.

[8] Date, C. J. , "An introduction to database systems", Third edition , Narosa publishing company, 1985.

[9] Finch, B. J. , and Cox, J. F. , "Strategic use of WIP inventory: The impact of bill-of-material shape and Plant-type", Production and inventory management journal-first quarter, 1989.

[10] Gessner, R. A. , "Master Production Scheduling planning" , John Wiley & sons, 1986.

[11] Gunder, W. , "Bill of material structuring: Modular and phantom bills", Production and inventory management journal-second quarter, 1978.

[12] King, B. E. , and Benton, W. C. , "Master production scheduling, customer service and manufacturing flexibility in an assemble-to-order environment", Int. J. Prod. Res. , Vol. 26, No. 6, pp. 1015-1036, 1988.

[13] Mather, H. , "Design , bills of material, and forecasting-The inseparable threesome", Production and inventory management journal, Vol. 27, No. 1, pp. 90-107, 1986.

[14] Nandakumar, G. , "Bills of material processing with a SQL database", Computers and Industrial engineering, Vol. 18, No. 4, pp. 471-483, 1990.

[15] Orlicky, J. , "Material requirement planning", McGraw-Hill Book company 1975.

[16] Panisset, B. D. , "MRP II for repair and refurbish industries", Production and inventory management journal -fourth quarter, 1988.

[17] Plossl, G. W. , "Production and inventory control principles and techniques", Prentice-Hall of India, 1985.

[18] Ross, D. F. , "The role of information in implementing MRP II systems", Production and inventory management journal-third quarter, 1989.

[19] Stultz, R. A. , "Illustrated dBASE IV", BPB Publications.

[20] Subhrahmanyam, R. V. S. ,M-tech thesis entitled " Management information system in a FMS and CIMS", IIT, Kanpur 1990.

[21] Tersine, R. J. , " Principles of inventory and materials management", North-Holland, 1988.

[22] User's manual on dBASE IV, Ashton-Tate Corporation, 1990.

[23] Vollmann,T. E. , Berry, W. L. , Whybark, D. C. , "Manufacturing planning and control systems", Galgotia Publications, 1989.

[24] Wight, O. W. , "Production and inventory management in the computer age, CBI publishing company, Inc, Boston, 1974.

[25] Wortmann, J. C. , Remmer, F. , and De Brock, E. O. , " A conceptual model for product-types and bills-of-materials: A set-theoretic approach", Proceedings of the IFIP TCS/WG 57 - Databases for production management, North-Holland, 1990.

[26] Wortmann, J. C. , and Van veen, E. A. , "Generic bills of materials in an assemble-to-order manufacturing", Int. J. Prod.Res. , Vol. 25, No. 11, pp. 1645-1658, 1987.

APPENDIX A

LIST OF DATABASE FILES AND THEIR STRUCTURE

The various relations that have been created to achieve the implementation have been grouped into the following categories:

Features

Structures

Equipment

Inventory

Overhaul

The various relations that together constitute these are listed below showing the entity, attributes and the key fields. The file name is shown at the left extreme and the detailed description given alongside. The structure of the file follows this. The width of the field are given within enclosed brackets alongside the datatype of the field. All files have the standard ".dbf" extension.

Features category

field_name	description	datatype
1.	End_equipment features relations	
normalf	features database for normal category of the equipment(combat vehicle).	
iden_no	equipment model number	character(4)
fording_ty	fording type	character(10)
calibre_mg	calibre of main gun (in mm)	character(5)
night_v_de	type of night vision device	character(10)
steer_type	type of steering mechanism	character(10)
amn_st_cap	ammunition storage capacity	numeric(2)
ditch_xing	ditch crossing capability(in m)	character(3)
aux_engine	whether auxillary fitted or not	character(3)
pre_heater	whether pre_heater fitted or not	character(3)
anti_air_c	type of anti aircraft weapon fitted	character(10)

tot_weight	total wt of the combat vehicle(in tons)	numeric(2)
gen_remarks	general remarks that cannot be listed under any specific feature	remarks(100)
key field	iden_no	

note: Similar relations are available for each master category of the end_equipment. For e.g. , in our case the recovery and bridge category of end_equipment.

2. Assembly features relations

egaf Features database for the engine assembly.

assy_no	model number of assembly	character(6)
disp_volum	displacement volume of engine(in cubic centimeter)	character(10)
horse_power	horse power rating of engine	character(5)
weight	total weight of engine in kilograms	character(6)
height	total height in millimeters	character(6)
cooling_typ	type of cooling system	character(10)
lubric_typ	type of lubrication system	character(10)
gen_remark	general remarks about the model	character(100)
key field:	assy_no	

note: similar relations are available for each type or family of assembly. For e.g. , in our case they are main gun assembly, bridge assembly, winch gear box assembly, etc..

3. Subassembly features relations

acsuaf Features database for the air_compressor subassembly

suas_no	subassembly model number	character(6)
wateroilse	water oil separator fitted or not	character(3)
output	output rating of compressor(in kg/cm^2)	character(10)
drive	type of drive	character(10)
gen_remark	general remarks about the model	character(100)
key field:	suas_no	

note: similar relations are available for the various

subassemblies. For e.g. , in our case they are, voltage regulator, fuel injection pump(FIP), etc..

Structures category

1. End_equipment -> assemblies relations.

normals structures database of the "normal" category of the equipment.

iden_no	individual model number	character(4)
engine	type of engine assembly fitted	character(6)
gear_box	type of gear box assembly fitted	character(6)
interg_box	type of intermediate gear box	character(6)
laserr_fin	type of laser range finder	character(6)
sus_system	type of suspension system	character(6)
main_gun	type of main gun fitted	character(6)
item_gr_00	K-number for items that go into the end_equipment and belonging to the item master group "prime_mover"	character(10)
item_gr_01	same as above for group "communication system"	character(10)
item_gr_02	same as above for group "weapon system"	character(10)
item_gr_03	same as above for group "vision and range devices"	character(10)
item_gr_06	same as above for group "controls and instrumentation"	character(10)
item_gr-07	same as above for group "accessories and mountings"	character(10)
item_gr_08	same as above for group "miscellaneous internal"	character(10)
item_gr_09	same as above for group "miscellaneous external"	character(10)

key field: iden_no

note: Similar relations are available for the other master categories of end_equipment. For e.g. , in our case they are

recovrs for the recovery category and bridges for the bridge category.

2. Assembly -> subassemblies relation

egas structures database of the assembly type "engine".

assy_no	assembly model number	character(6)
fuel_in_pu	type of FIP fitted	character(6)
air_compre	type of air_compressor fitted	character(6)
cyl_block	type of cylinder block fitted	character(6)
auxy_drive	type of auxillary drive mechanism	character(6)
item_gr_00	K-number for those loose items fitted on the engine assembly and belonging to the item master group "prime_mover"	character(10)
item_gr-07	same as above for item master group "accessories and mountings"	character(10)
item_gr-08	same as above for group "miscellaneous internal"	character(10)

key field: assy_no

note: similar relations are available for each type of the assembly. For e.g. , in our case they are, mgas, for the main gun assembly, wbas, for the winch gear box assembly, etc..

3. Subassembly->items relations

acsuas structure relation for the subassembly air compressor.

suas_no	subassembly model number	character(6)
separator	type of oil separator fitted	character(6)
pump_shaft	type of pump shaft fitted	character(6)
filter	model of filter fitted	character(6)
reducer	model or type of reducer fitted	character(6)
item-gr-00	K-number for the loose items that go into this subassembly and belong to item master group"prime_mover"	character(10)

key field: suas_no

note: similar relations are available for the other subaasemblies also. For e.g. , in our case they are vosuas, for the voltage regulator subassembly, fisuas, for the FIP, etc..

4. **Grouping of items**

The grouping of items has been done on a system basis i.e. , for example all items that are used in the communication subassemblies and assemblies such as radio sets, antenna assembly, are grouped under the communication system category. The various item groups in our case are:(the group names are self explanatory and the grouping itself is only demonstrative and by no means exhaustive)

group name	group code
prime_mover	00
communication system	01
weapon system	02
vision and range devices	03
bridging	04
recovery	05
controls and instrumentation	06
accessories and mountings	07
miscellaneous internal	08
miscellaneous external	09

5. BOM files One BOM file for each item master group is available to capture the parent_component relationship of all assemblies and subassemblies in that group.

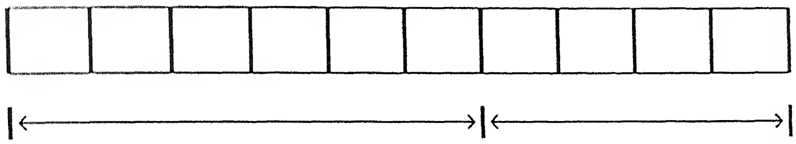
prbom BOM file for master group "prime_mover"

item	individual unique identity of item	character(10)
parent	individual unique identity of parent	character(10)
quantity	quantity of item required by parent	numeric(6)
key fields:	item within parent for product structure explosion	
	parent within item for product structure implosion	

note: (1) Similar BOM files are available for each item group

(2) Although the item identity code is only 6 characters

wide the width of item and parent fields have been kept at 10 accommodate for K-numbers and the items or assemblies that are used in the overhaul of various assemblies. The K-numbers and items used for overhaul are identified as follows:



standard item identifier as used for all items	extension " -999 " used for K-numbers and " -888 " used for overhaul or repair kit.

Inventory category

1. source relations

prsource source relation for item group prime_mover

item_no	individual item number	character(6)
vendor_no	unique vendor number	character(5)
unit_price	unit price of this item with the vendor	numeric (10)
s_lead_time	supply lead time required by vendor in weeks	numeric(2)

key field: vendor_no within item_no
item_no within vendor_no

note: similar relations are available for each of the item group

2. vendor relation

vendor this has the information about the details of the vendors

vendor_no	vendor number	character(5)
vendor_nam	vendor name	character(15)
street		character(20)
city		character(10)

state		character(10)
pin_code		character(6)
vendor_rat	vendor rating	character(1)

key field: vendor_no

Overhaul category

1. assembly repair kit relations

egrepkit repair kit database for the engine assembly

assy_no	assembly model number	character(6)
rep_cat_no	repair category number	character(1)
labor_hrs	labor hours required to complete overhaul	
fuel_in_pu	kit for fuel injection pump	character(10)
air_compre	kit for overhaul of air compressor	character(10)
cyl_block	kit for overhaul of cylinder block	character(10)
auxy_drive	kit for auxillary drive sub assy	character(10)
item_gr_00	k-number for item group prime mover	character(10)
item_gr_07	K-number for item group accessories and mountings	character(10)
item_gr_08	K-number for item group miscellaneous internal	character(10)

key_field: repair category within assembly model number

note: similar relations are available for the different assemblies. For example in our case brrepkit for the bridge assembly, etc..

2. resources utilized relation

asrepwk resources required for repair of assembly

assy_no	assembly model number	character(6)
rep_cat_no	repair category number	character(1)
wkcenter01	time required on work center number "01" for repair of assembly	numeric(3)

wkcenter02	time required on work center number "02" for repair of assembly with repair category rep_cat_no	numeric(3)
remarks	general remarks about overhaul of assembly model falling under a specific repair category	character(100)

key_field: repair category within assembly model number

note: The number of work centers shown are not exhaustive

In_service Equipment category

1. In_service Equipment constituents relations

normalf In_service equipment constituents database for normal category of equipment.

iden_no	unique equipment identity number	character(10)
engine	model and number of engine fitted	character(10)
gear_box	model and serial number of gear box fitted in the end_equipment	character(10)
interg_box	model and serial number of intermediate box fitted in the end_equipment	character(10)
laserr_fin	model and serial number of laser range finder fitted	character(10)
remarks	generalremarks about specific equipment	character(100)

key_field: Unique equipment number

note: Similar relations exist for the different categories of the end_equipment.

2. Equipment performance history relations

normalp performance history database for equipment category normal.

iden_no	unique equipment number	character(10)
kms_done	total kilometers done	numeric(4)
hours_run	total hours run of the equipment	numeric(4)
no_off_eng	whether first,second or third engine	

	fitted	numeric(1)
no_off_oh	number of overhauls undergone earlier	numeric(1)
gen_remark	general remarks about the individual equipment to include information that cannot otherwise br listed under any fields.	character(100)

key_field : iden_no

note: Similar relations are available for each of the other end_equipment categories.

APPENDIX-B

LIST OF PROGRAMS

The list of programs with their functions in brief are listed menu wise below. All program files have an extension of ".prg" .

No	name	function
I	Main Menu	
1.	mainmenu	Responsible for main menu generation
II	Features Submenu	
1.	modfeat	Equipment_model_no -> features
2 .	featmod	Given a list of features->equipment_model_No
3.	assyfeat	Assembly_model_No -> features
4.	featassy	Given a list of features -> assembly_model_No
5.	subafeat	Subassembly_model_No -> features
6.	featsuba	Given a list of features->subassembly_model_no
7.	featstru	Structure of the model of equipment chosen by program No. 2 above
8.	feasstru	Structure of the model of assembly chosen by program No. 4 above
9.	fesustru	Structure of the model of subassembly chosen by program No. 6 above
III	Structures Submenu	
1.	modstru	Equipment_model_no -> structure
2.	assystru	Assembly_model_no -> structure
3.	subastru	Subassembly_model_no -> structure
4.	peggstru	Item_no -> implosion details
5.	master	product explosion
6.	master31	summarized explosion
7.	final31 and output31	for level multiplication and addition in summarized explosion

IV In_service Equipment Management Submenu

1.	eqptcons	Equipment_no -> constituents
2.	eqptperf	Equipment_no -> performance characteristics
3.	perfeqpt	Performance characteristics -> equipment_no of those with these characteristics in service
4.	eqptfeat	Equipment_no -> features
5.	feateqpt	Features list -> equipment _nos of those equipment in service possessing these features

V Inventory Management Submenu

1.	itemaval	Item_no -> availability status
2.	iteminfo	Item_no -> general information about the item
3.	itemused	Item_no -> where and how much used information about an item
4.	itemordr	Selective and exhaustive ordering of items

VI Overhaul Management Submenu

1.	eqptover	Equipment_no -> complete requirement of items to carry out overhaul of the equipment
2.	assyover	Assembly_no &repair category->complete requirement of items to carry out the overhaul of the assembly
3.	itemover	Item_no -> used for overhaul of which all assemblies information
4.	rescused	Assembly_no & repair_category -> resources utilized

VII Editing Operations Submenu

1.	adder	Addition of a record to any relation
2.	editor	Editing the record of any database
3.	deleter	Delete the record of any database

APPENDIX C

INTER-RELATIONSHIP BETWEEN PROGRAM FILES AND DATABASE FILES

Having seen the details of the database files and the program files, we now proceed to show the interrelationship between them. This is necessary to justify the creation and maintenance of the various data files. The interrelationship between the various database or files with ".dbf" extension and the program files or those files with ".prg" extension are shown database wise in this appendix. The methodology adopted is to list the files of a database on the left side inside a box and list the various programs that use these datafiles module wise on the right.

FEATURES DATABASE

Structures Submenu

1. modfeat.prg
2. featmod.prg
3. assyfeat.prg
4. featassy.prg
5. subafeat.prg
6. featsuba.prg

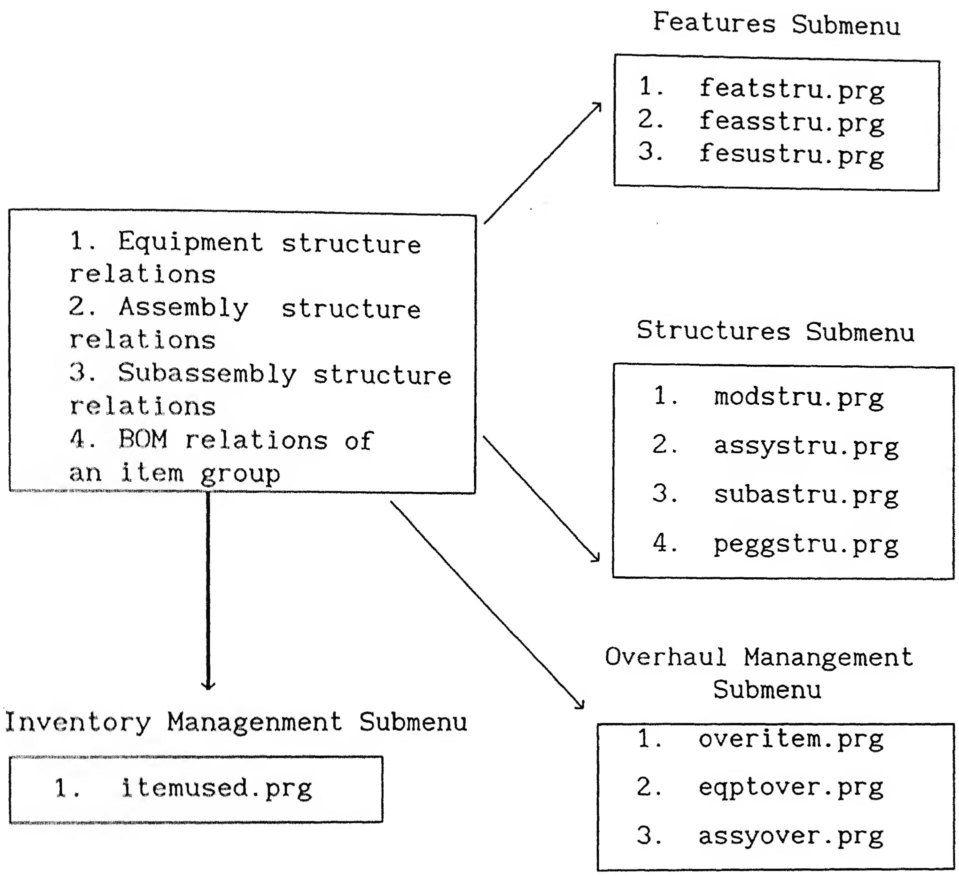
1. End_Equipment features relations
2. Assembly features relations
3. Subassembly features relations

In_service Equipment Submenu

1. eqptfeat.prg
2. feateqpt.prg

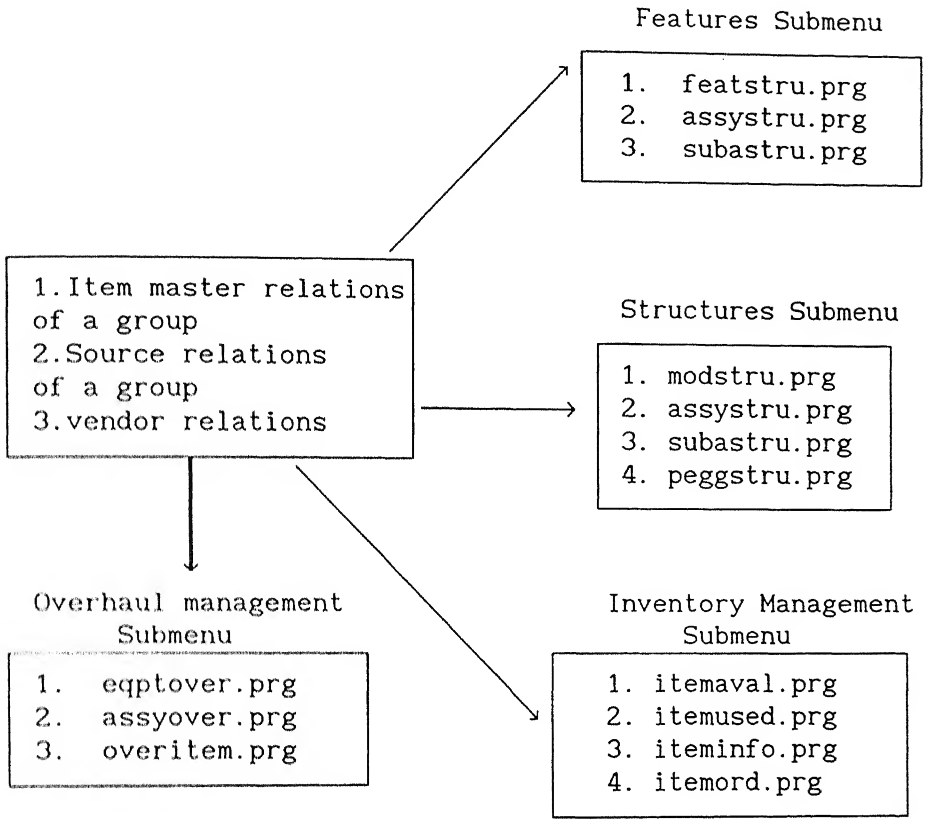
note: These relations store information about the features of the various models of the end_equipment, assembly, subassembly as the case may be, and are primarily meant for selection of a model of an equipment, assembly, or subassembly, and to list the features of any model of end_equipment, assembly, or subassembly that is being manufactured.

STRUCTURES DATABASE



note: These relations store information about the immediate lower level components that go into into a model of an end_equipment, assembly, or subassembly as the case may be. These components may be an assembly in the end_equipment structure relation, or a subassembly in the assembly structure relation. This information is extracted during the product structure explosion of the end_equipment, assembly, or the subassembly as the case may be. The BOM relations of each item group stores the complete parent-component relationship of every assembled item in a group.

INVENTORY DATABASE



Note: The inventory management database has the complete information about the complete inventory. This information would include, safety stock of an item, quantity on hand, item description, etc.. The source relations of an item group store the information about the vendors who can supply an item, their supply lead time and their price per item. The vendor relations in turn store the complete information about the vendor itself which includes, address of the vendor and the rating of the vendor.

IN_SERVICE EQUIPMENT DATABASE

1. Equipment constituents relations
2. Equipment performance history relations

→

Equipment Management Submenu

1. eqptcons.prg
2. eqptperf.prg
3. perfeqpt.prg
4. eqptfeat.prg
5. feateqpt.prg

Note: The equipment constituent relations stores the information the various models and the serial number of the assemblies fitted in the end_equipment. The equipment performance history relations contains information about the usage or in_service exploitation data about an equipment.

OVERHAUL DATABASE

1. Assembly-repair kit relaions
2. Assembly, repair category-resources utilized relations
3. Work center details relations

→

Overhaul Management Submenu

1. eqptover.prg
2. assyover.prg
3. rescused.prg

Note: The assembly model number repair category relations contains the information about the repair kit or the items that may be required to carry out the overhaul of the assembly under a certain repair category. The resource utilized relations contains the information about the hours required on a work center by an assembly with a certain repair category. The work center details relations contains the description of the work center themselves.

APPENDIX D

SOME SAMPLE OUTPUTS

Sample outputs for product structure explosion, indented and summarized are shown in the succeeding pages. The indented and summarized explosion are shown for a model of the end_equipment belonging to the normal category. The enclosures are the exact replica of the actual output generated by the system. This output being for an end_equipment is demonstrative of proceeding downwards level by level as also the recursive explosion from the subassembly level downwards. A "$" in the output indicates that there are no items below this parent. The product structure shown is only demonstrative and not exhaustive. Although in practice all the K-numbers will have a number of items going into it, the same has not been shown in some cases.

SAMPLE INDENTED EXPLOSION FOR A MODEL OF END_EQUIPMENT

Sample output for a model of end_equipment belonging to the normal category of combat vehicle is shown below. The output that was written onto a text file by the explosion program of dBASE IV, is shown here.

THE LEVEL ONE ASSEMBLIES OF THE CHOSEN EQUIPMENT ARE
OF CATEGORY NORMAL ARE

IDEN_NO	0100
ENGINE	eg0001
GEAR_BOX	gb0001
INTERG_BOX	ig0001
LASERR_FIN	lr3001
SUS_SYSTEM	ss0001
MAIN_GUN	mg2001
ITEM_GR_00	pr0001-999
ITEM_GR_01	cn1001-999
ITEM_GR_02	ws2001-999
ITEM_GR_03	vr3001-999
ITEM_GR_06	ci6001-999
ITEM_GR_07	am7001-999
ITEM_GR_08	mi8001-999
ITEM_GR_09	me9001-999

THE LEVEL ONE DOWN TO TWO OR THE SUBASSEMBLIES OF
OF ENGINE ARE AS FOLLOWS

ASSY_NO	eg0001
FUEL_IN_PU	fi0001
AIR_COMPRE	ac0001
CYL_BLOCK	cy0001
AUXY_DRIVE	ad0001
ITEM_GR_00	pr0002-999
ITEM_GR_07	am7002-999
ITEM_GR_08	mi8002-999

THE STRUCTURE OF EACH OF THE SUBASSEMBLIES OF
OF ENGINE OF THE EQPT ARE AS FOLLOWS

THE DOWNWARDS EXPLOSION OF FUEL_IN_PU OF ENGINE OF THE EQUIPMENT
IS AS FOLLOWS

ITEM	PARENT	QUANTITY
fp0001	fi0001	1
in0001	fi0001	6
in0100	fi0001	6
pb0100	fp0001	1
sr0010	fp0001	3
wh0001	fp0001	6
ft0001	in0001	1

in0300	in0001	1
nu0001	in0001	3
sr0001	in0001	2
bb0001	in0100	1
bh0001	in0100	1
pb0001	in0100	2
ps0001	in0100	2
$	pb0100	0
$	sr0010	0
$	wh0001	0
$	ft0001	0
$	in0300	0
$	nu0001	0
$	sr0001	0
$	bb0001	0
$	bh0001	0
$	pb0001	0
$	ps0001	0

THE DOWNWARDS EXPLOSION OF AIR_COMPRE OF ENGINE OF THE EQUIPMENT
IS AS FOLLOWS

ITEM	PARENT	QUANTITY
ar0001	ac0001	3
as0001	ac0001	2
ft0002	ac0001	2
pr0005-999	ac0001	1
ps0010	ac0001	3
cl0001	ar0001	5
ru0001	ar0001	4
co0001	as0001	4
pi0001	as0001	3
pi0002	as0001	4
$	ft0002	0
$	pr0005-999	0
$	ps0010	0
$	cl0001	0
$	ru0001	0
sr0010	co0001	5
bh0001	pi0001	3
bb0001	pi0002	6
$	sr0010	0
$	bh0001	0
$	bb0001	0

THE DOWNWARDS EXPLOSION OF CYL_BLOCK OF ENGINE OF THE EQUIPMENT
IS AS FOLLOWS

ITEM	PARENT	QUANTITY
pl0001	cy0001	7
$	pl0001	0

THE DOWNWARDS EXPLOSION OF AUXY_DRIVE OF ENGINE OF THE EQUIPMENT
IS AS FOLLOWS

ITEM	PARENT	QUANTITY
nu0003	ad0001	3
pi0003	ad0001	4
$	nu0003	0
$	pi0003	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_00 OF ENGINE OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
bo0001	pr0002-999	10
cm0001	pr0002-999	5
$	bo0001	0
$	cm0001	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_07 OF ENGINE OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
$	am7002-999	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_08 OF ENGINE OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
$	mi8002-999	0

THE LEVEL ONE DOWN TO TWO OR THE SUBASSEMBLIES OF OF GEAR_BOX ARE AS FOLLOWS

ASSY_NO	gb0001
CONTROLLER	gc0001
CASING	gs0001
SYNCHROMES	gm0001
ITEM_GR_00	pr0003-999
ITEM_GR_07	am7003-999
ITEM_GR_08	mi8003-999

THE STRUCTURE OF EACH OF THE SUBASSEMBLIES OF OF GEAR_BOX OF THE EQPT ARE AS FOLLOWS

THE DOWNWARDS EXPLOSION OF CONTROLLER OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
cb0001	gc0001	2
gs0300	gc0001	1
ot0001	gc0001	3
pr0002-999	gc0001	1
$	cb0001	0
$	gs0300	0
$	ot0001	0
bo0001	pr0002-999	10
cm0001	pr0002-999	5
$	bo0001	0

$	cm0001	0

THE DOWNWARDS EXPLOSION OF CASING OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
gr0001	gs0001	5
nu0003	gr0001	4
$	nu0003	0

THE DOWNWARDS EXPLOSION OF SYNCHROMES OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
bo0001	gm0001	10
nu0003	gm0001	5
$	bo0001	0
$	nu0003	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_00 OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
nu0002	pr0003-999	8
pi0003	pr0003-999	5
$	nu0002	3
$	pi0003	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_07 OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
$	am7003-999	0

THE DOWNWARDS EXPLOSION OF ITEM_GR_08 OF GEAR_BOX OF THE EQUIPMENT IS AS FOLLOWS

ITEM	PARENT	QUANTITY
$	mi8003-999	0

Note: The sample output is shown for two of the assemblies of the chosen model of the master category of equipment only. The remaining are similar and are not listed for brevity's sake. As indicated earlier in appx A a "$" denotes that there is no item or items below the parent against which it is shown, consequently the quantity against such an entry will be a "0". The sample output has been shown for the engine assembly and the gear box assembly of the end_equipment. The structure is representative only and not exhaustive.

SAMPLE OUTPUT FOR SUMMARIZED EXPLOSION

Note: The summarized explosion of the model of the end_equipment for which the product structure (or indented explosion) was shown earlier in this appx, is now shown here. The items have been listed in the alphabetic order of their identities. The quantity shown are the total quantities of the item required for the assembly of the end_equipment.

THE SUMMARIZED EXPLOSION OF THE PRODUCT/ITEM IS AS FOLLOWS

ITEM	ITEM DESCRIPTION	QUANTITY
an1000	ANTENNA	75
an1200	ANTENNA	75
ar0001	AIR REDUCER	3
as0001	WATER_SEPARATOR	2
ba2001	BREECH ASSEMBLY	5
bb0001	BALL BEARING	54
bd0001	BODY GEAR	12
bh0001	BALL BEARING HOUSING	24
bh2001	BREECH HOLDER	3
bm2001	BREECH MECHANISM	10
bo0001	BOLT	45
bo0800	BOLT	1200
bo0900	BOLT ACTION	90
bo2001	BOLT	960
bx1000	RS123 SPARE BOX	1
cb0001	GEAR CONTROL BOX	2
cl0001	CLIP	15
cl1000	CLAMP	420
cm0001	CLAMP	10
cn1000	CONNECTOR	15
cn1200	CONNECTOR	6
co0001	CONSOLE	8
co1000	CORD	15
co1200	CORD 1A	6
cy2001	CYLINDER	12
fe1000	FEEDER	1500
fp0001	FUEL FEED PUMP ASSY	1
ft0001	FILTER	6
ft0002	FILTER	2
gm2001	CO-AXIAL M/C GUN 7.6	15
gm2002	CO-AXIAL M/C GUN 14.	7
gr0001	GEAR	5
gr0003	GEAR	3
gr0004	GEAR	3
gs0300	GEAR CONTROLLER CSNG	1
ho1000	HOUSING	150
ho1030	HOUSING	21600
ho2001	HOUSING	24
in0001	INJECTOR ASSEMBLY	6

in0100	INJECTOR PUMP ASSY	6
in0300	INJECTOR BODY	6
ip1000	INTERPHONE 124	3
ip1003	INTERPHONE 124A	5
li1000	LINE WITH PROTECTOR	30
md2001	MANDREL	5
mt0001	SPECIAL MOUNTING	100
nu0001	NUT	24038
nu0002	NUT	8
nu0003	NUT	12053
nu0900	CASTELLATED NUT	1000
nu2001	NUT	2970
ot0001	OIL THROWER	3
pb0001	PUMP BODY	12
pb0100	PUMP BODY UNIT	1
pg0001	PACKING GLAND	5
pi0001	PIPE	6
pi0002	PIPE	8
pi0003	PIPE	360009
pl0001	COVER PLATE ENGINE	7
pr0002-999		1
pr0005-999		1
ps0001	PUMP SHAFT	12
ps0010	PUMP SHAFT	3
rg1000	RING	375
rs1000	RADIO SET 123	3
rs1100	RADIO SET 113	3
ru0001	REGULATOR AIR	12
sl2001	SLEEVE	30
so1000	SOCKET	240
sr0001	SEALING RING	17
sr0010	SEALING RING	43
st0002	STRIP STEEL	12000
st1000	STOPPER	30
st2001	STRIP	310
sw1000	SWITCH	150
tl1000	TERMINAL	375
wh0001	WASHER	240006
wh0303	WASHER	60
wh1000	WASHER	150
wi0001	ZINCED WIRE	500

Note: The items with an extension "-999" against their numbers denote K-numbers(or K-bills) and as such do not have a description tag for themselves.